

Moduł szkoleniowy PHP

PHP: Hypertext Preprocessor

poziom średnio zaawansowany 15 godzinny

Spis treści

1	Metryka dokumentu.....	5
2	Cel.....	6
3	Opis sposobu realizacji celów.....	6
4	Treści kształcenia.....	6
5	Opis założonych osiągnięć ucznia.....	6
6	Sposoby osiągania celów.....	7
7	Propozycje kryteriów oceny i metod sprawdzania osiągnięć ucznia	7
8	Test końcowy sprawdzający wiedzę.....	7
9	Lekcje.....	9
9.1	Lekcja 1 - Architektura usług sieciowych (Klient – serwer, Peer to peer, Architektura hybrydowa).....	9
9.1.1	Cel lekcji.....	9
9.1.2	Treść - slajdy z opisem	9
9.1.3	Ćwiczenia.....	18
9.1.4	Opis założonych osiągnięć ucznia.....	18
9.2	Lekcja 2 - Protokół http (budowa protokołu HTTP 1.1, nagłówki żądania i odpowiedzi, komunikacja http)	18
9.2.1	Cel lekcji.....	18
9.2.2	Treść - slajdy z opisem	19
9.2.3	Ćwiczenia.....	27
9.2.4	Opis założonych osiągnięć ucznia.....	27
9.3	Lekcja 3 - Modyfikacja nagłówków http (Lista nagłówków, Wykorzystanie funkcji header, Zastosowania w skryptach PHP).....	27
9.3.1	Cel lekcji.....	27
9.3.2	Treść - slajdy z opisem	27
9.3.3	Ćwiczenia.....	42
9.3.4	Opis założonych osiągnięć ucznia.....	42
9.4	Lekcja 4 – Cookie - informacje o użytkowniku (Budowa i działanie cookie, Używanie cookie w aplikacjach, Definiowanie i obsługa z poziomu PHP, Bezpieczeństwo).....	42
9.4.1	Cel lekcji.....	42
9.4.2	Treść - slajdy z opisem	43
9.4.3	Ćwiczenia.....	50

9.4.4	Opis założonych osiągnięć ucznia	50
9.5	Lekcja 5 - Sesje - współdzielenie informacji (Struktura i sposób działania, Kontrola sesji w PHP, Wykorzystanie sesji w aplikacjach internetowych)	50
9.5.1	Cel lekcji.....	50
9.5.2	Treść - slajdy z opisem	51
9.5.3	Ćwiczenia	59
9.5.4	Opis założonych osiągnięć ucznia	59
9.6	Lekcja 6 - Poczta - systemy SMTP (Systemy pocztowe w Internecie, Budowa protokołu SMTP, Komunikacja SMTP, Struktura nagłówków wiadomości, Obsługa poczty w aplikacjach PHP)	59
9.6.1	Cel lekcji.....	59
9.6.2	Treść - slajdy z opisem	59
9.6.3	Ćwiczenia	69
9.6.4	Opis założonych osiągnięć ucznia	70
9.7	Lekcja 7 - MIME - rozszerzenia multimedialne (Struktura nagłówków MIME, Kodowanie Base64 i Quoted-Printable, Wiadomości wieloczęściowe, Obsługa poczty multimedialnej w PHP)	70
9.7.1	Cel lekcji.....	70
9.7.2	Treść - slajdy z opisem	70
9.7.3	Ćwiczenia	79
9.7.4	Opis założonych osiągnięć ucznia	79
9.8	Lekcja 8 - UML 2.1 (Diagramy czynności, Projektowanie złożonych aplikacji)	79
9.8.1	Cel lekcji.....	79
9.8.2	Treść - slajdy z opisem	79
9.8.3	Ćwiczenia	89
9.8.4	Opis założonych osiągnięć ucznia	90
9.9	Lekcja 9 - Bezpieczeństwo aplikacji (Rodzaje ataków na portale internetowe, Testowanie poszczególnych zagrożeń, Zabezpieczanie aplikacji)	90
9.9.1	Cel lekcji.....	90
9.9.2	Treść - slajdy z opisem	90
9.9.3	Ćwiczenia	102
9.9.4	Opis założonych osiągnięć ucznia	103
9.10	Lekcja 10 - Dobre praktyki programowania w PHP	103
9.10.1	Cel lekcji.....	103
9.10.2	Treść - slajdy z opisem	103

Moduł szkoleniowy SQL - Structured Query Language str. 3

Człowiek - najlepsza inwestycja

9.10.3	Ćwiczenia.....	111
9.10.4	Opis założonych osiągnięć ucznia.....	111

1 Metryka dokumentu

Szczeciński Park Naukowo Technologiczny			
Dokument	Moduł Szkoleniowy PHP: Hypertext Preprocessor		
Krótki opis dokumentu	Moduł szkoleniowy kursu PHP zawierający opis celów, treści poszczególnych lekcji, ćwiczeń oraz test końcowy sprawdzający wiedzę		
Data druku	12.02.2013	Liczba stron	111
Nazwa pliku	PHP - Moduł szkoleniowy.docx	Status	roboczy

Historia zmian

Nr wersji	Data	Opis	Działanie (*)	Autorzy
0.1	01.02.2013	Utworzenie nowego dokumentu	N	Bartosz Komin, Daily Group Sp. z o.o.
1.0	12.02.2013	Wersja gotowa do publikacji	Z	Bartosz Komin, Daily Group Sp. z o.o.

(*) Działanie: N-Nowy, Z-Zmiana, W-Weryfikacja

Lista dystrybucyjna

Imię i nazwisko / Rola	Organizacja

Zgłoszono do odbioru

Imię i nazwisko		Data:		Podpis:	
Imię i nazwisko	Bartosz Komin	Data:	12.02.2013	Podpis:	

2 Cel

Poznanie zagadnień z zakresu języka PHP oraz protokołów sieciowych takich jak HTTP, SMTP oraz MIME. Następnie wprowadzenie do mechanizmów ciasteczek i sesji. Zdobycie wiedzy na temat podstawowych architektur aplikacji internetowych oraz bezpieczeństwa stron, bądź portali internetowych. Zobrazowanie jak radzić sobie z atakami na strony, przez wykorzystanie podstawowych metod zabezpieczeń. Kurs ma również na celu zachęcić uczestnika do dalszego rozwoju, poprzez wskazanie mu bibliotek, frameworków raz narzędzi upłynniających i ułatwiających pracę z językiem PHP.

3 Opis sposobu realizacji celów

10 półtoragodzinnych lekcji składających się przedstawienia materiału wraz z przykładami i ćwiczeń praktycznych. Po zakończeniu całego cyklu - przeprowadzenie egzaminu sprawdzającego wiedzę.

4 Treści kształcenia

Treść kursu została podzielona na 10 bloków tematycznych – po jednym do każdej lekcji:

1. Architektura usług sieciowych (Klient – serwer, Peer to peer, Architektura hybrydowa);
2. Protokół http (budowa protokołu HTTP 1.1, nagłówki żądania i odpowiedzi, komunikacja http);
3. Modyfikacja nagłówków http (Lista nagłówków, Wykorzystanie funkcji header, Zastosowania w skryptach PHP);
4. Cookie - informacje o użytkowniku (Budowa i działanie cookie, Używanie cookie w aplikacjach, Definiowanie i obsługa z poziomu PHP, Bezpieczeństwo);
5. Sesje - współdzielenie informacji (Struktura i sposób działania, Kontrola sesji w PHP, Wykorzystanie sesji w aplikacjach internetowych);
6. Poczta - systemy SMTP (Systemy pocztowe w Internecie, Budowa protokołu SMTP, Komunikacja SMTP, Struktura nagłówków wiadomości, Obsługa poczty w aplikacjach PHP);
7. MIME - rozszerzenia multimedialne (Struktura nagłówków MIME, Kodowanie Base64 i Quoted-Printable, Wiadomości wieloczęściowe, Obsługa poczty multimedialnej w PHP);
8. UML 2.1 (Diagramy czynności, Projektowanie złożonych aplikacji);
9. Bezpieczeństwo aplikacji (Rodzaje ataków na portale internetowe, Testowanie poszczególnych zagrożeń, Zabezpieczanie aplikacji).
10. Dobre praktyki programowania w PHP

5 Opis założonych osiągnięć ucznia

Uczeń po odbyciu kursu pozna protokoły sieciowe, mechanizm ciasteczek i sesji, nauczy się zabezpieczać swoje aplikacje, pozna podstawowe architektury aplikacji internetowych i nauczy się je projektować, będzie posiadał wytyczoną ścieżkę dalszego rozwoju.

6 Sposoby osiągnięcia celów

Podczas procesu kształcenia, uczeń powinien wykonywać ćwiczenia oraz przykłady zawarte w materiałach edukacyjnych. Po ukończeniu szkolenia, każdy uczestnik musi zaliczyć odpowiedni test, który pomoże zweryfikować jego świeżo nabytą wiedzę oraz określić ewentualne luki w wiedzy, który powinien uzupełnić.

7 Propozycje kryteriów oceny i metod sprawdzania osiągnięć ucznia

Ocena zaliczenia przedmiotu powinna być składową dwóch ocen: testu oraz ocen częściowych za ćwiczenia :

Kryteria oceny testu:

- 0-7 poprawnych odpowiedzi - 1
- 8-9 poprawnych odpowiedzi - 2
- 10-11 poprawnych odpowiedzi - 3
- 12-13 poprawnych odpowiedzi - 4
- 14-15 poprawnych odpowiedzi - 5

Kryteria oceny ćwiczeń:

- Rozwiązania ćwiczeń spełniają podstawową funkcjonalność - 3
- Rozwiązania ćwiczeń spełniają podstawową funkcjonalność oraz działają poprawnie w każdym przypadku - 4
- Spełniają powyższe wymagania oraz są zabezpieczone na błędne dane - 5

8 Test końcowy sprawdzający wiedzę

15 zadań na 30 minutowy test sprawdzający wiedzę. Pogrubioną czcionką zaznaczono prawidłowe odpowiedzi.

1. Czym różni się usługa REST w stosunku od innych webservice'ów opartych o WSDL:
 - a. Brak stanowości zasobów**
 - b. Posiada opis formatu zapytań i odpowiedzi
 - c. Jest kompilowany
 - d. Wykorzystuje metody HTTP**
2. Która metoda z niżej wymienionych nie jest metodą HTTP w wersji 1.1:
 - a. CONNECT**
 - b. PUT
 - c. OPTIONS
 - d. TRACE
3. Jeżeli w nagłówku HTTP chcemy określić główny język dokumentu jako polski, stosujemy składnię:
 - a. Content-Language: en, pl**



- b. Content-Language: pl, en
 - c. **Content-Language: pl**
 - d. Content-Language: en, pl, de
4. Jak usunąć ciasteczko o nazwie 'hello' w języku php:
- a. **setcookie("hello", "", time()-3600);**
 - b. deletecookie("hello");
 - c. unset("hello");
 - d. unsetcookie("hello");
5. Czy możemy ustawić zmienną w sessji przed wywołaniem funkcji session_start:
- a. Tak, za pomocą tablicy \$_SESSION;
 - b. Możemy, ale musimy użyć specjalnej funkcji;
 - c. **Nie;**
6. Podaj numer domyślnego portu SMTP?
- Odp: Port o numerze 25.
7. Kodowania transportowe to:
- a. **Base64**
 - b. **Quoted-Printable**
 - c. ISO 8859-2
 - d. Utf-16
8. Do diagramów języka UML nie zaliczamy:
- a. Diagramu klas
 - b. Diagramu pakietów
 - c. **Diagramu konstelacji**
 - d. Diagramu czynności
9. Rodzajem bezpośredniego ataku na portale internetowe lub ich serwery nie jest:
- a. PHP Injection
 - b. XSS Injection
 - c. SQL Injection
 - d. **Phishing**
10. Szkieletem aplikacji języka PHP nie jest:
- a. Zend Framework
 - b. Symfony
 - c. **Composer**
 - d. Yii
11. Do pisania jakich systemów służy język php:
- a. **Systemów działających po stronie serwera**
 - b. Aplikacji desktopowych
 - c. Systemów do obsługi serwera WWW
 - d. Sterowników urządzeń
12. Mechanizm sesji przechowuje dane (nie licząc identyfikatora) po stronie:
- a. Klienta
 - b. **Serwera**
 - c. Serwera oraz klienta

- d. W dowolnej bazie danych, wybranej przez użytkownika
 - e. Mechanizm sesji nie przechowuje żadnych danych
13. Który protokół z wymienionych poniżej, nie jest protokołem pocztowym:
- a. **FTP**
 - b. SMTP
 - c. POP3
 - d. IMAP
14. Do czego służy narzędzie phing:
- a. Jest to system zarządzania treścią
 - b. Generowania dokumentacji
 - c. **Automatyzacji instalacji w języku php**
 - d. Wysyłania poczty
15. Możliwą wartość type w nagłówku Content-Type protokołu MIME nie jest:
- a. text
 - b. application
 - c. **graphic**
 - d. audio

9 Lekcje

9.1 Lekcja 1 - Architektura usług sieciowych (Klient – serwer, Peer to peer, Architektura hybrydowa)

9.1.1 Cel lekcji

Celem lekcji jest wyjaśnienie podstawowych architektur aplikacji sieciowych: klient – serwer, p2p oraz architektury mieszanej. Nauczyciel opisuje i pokazuje ich schematy. Opis podstawowej architektury zorientowanej na usługi SOA. Opis jej najważniejszych implementacji SOAP, WebService, REST, DCOM, COBRA, WCF.

9.1.2 Treść - slajdy z opisem

Slide 1



Na lekcji zostaną omówione najczęściej spotykane architektury rozwiązań informatycznych, ich wady i zalety

W kolejnej części omówione będą tematy SOA i najczęściej używane architektury usług sieciowych



Slide 2

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP
Hypertext Preprocessor

Architektury rozwiązań informatycznych

- klient – serwer
- peer to peer
- hybrydowa

DAILY GROUP

Lekcja 1 – architektury rozwiązań informatycznych

Wyróżniamy architektury:

- klient - serwer - najbardziej popularna architektura oprogramowania w której klient (aplikacja kliencka) łączy się z serwerem (farmą serwerów),
- peer to peer - rozproszona architektura której założeniem jest możliwość połączenia się wszystkich aplikacji między sobą
- hybrydowa - szerokie pojęcie np. architektura mieszana klient - serwer + peer to peer.

Slide 3

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP
Hypertext Preprocessor

Architektury rozwiązań informatycznych

- klient – serwer
- peer to peer
- hybrydowa

DAILY GROUP

Lekcja 1 – architektury rozwiązań informatycznych

Wyróżniamy architektury:

- klient - serwer - najbardziej popularna architektura oprogramowania w której klient (aplikacja kliencka) łączy się z serwerem (farmą serwerów),
- peer to peer - rozproszona architektura której założeniem jest możliwość połączenia się wszystkich aplikacji między sobą
- hybrydowa - architektura mieszana serwer + peer to peer.

Slide 4

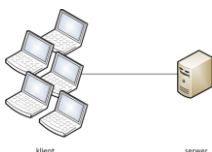
B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP
Hypertext Preprocessor

Architektury rozwiązań informatycznych

- klient – serwer



DAILY GROUP

Lekcja 1 – architektury rozwiązań informatycznych

Wyróżniamy architektury:

- klient - serwer - najbardziej popularna architektura oprogramowania w której klient (aplikacja kliencka) łączy się z serwerem (farmą serwerów),
- Przykładem jest serwer poczty elektronicznej (klient – np. outlook, serwer – exchange), serwery WWW (klient-przeglądarka, serwer-serwerWWW), serwery plików czy serwery aplikacji.

Stroną inicjującą komunikację jest klient. Stroną wykonującą usługę jest serwer.

W zależności od zastosowań wyróżnia się wiele typów architektury klient serwer:

- dwuwarstwowa (klient komunikuje się z serwerem np. serwer plików),
- trójwarstwowa (klient łączy się z serwerem aplikacji (warstwa druga) który przetwarza żądania, korzystając z baz danych (warstwa trzecia) lub innych zasobów odpowiada na żądanie,
- wielowarstwowa w której logika biznesowa została zaimplementowana na wielu poziomach.

Zalety: wspólne dane w jednym miejscu, zapewniona spójność i brak konfliktów wersji, łatwość zabezpieczenia danych

Wady: ograniczona wydajność (można stosować klastry wydajnościowe), awaria serwera powoduje niedostępność aplikacji/danych (stosuje się klastry niezawodnościowe),

Slide 5



B2E
 BUSINESS TO EDUCATION



SZCZECIŃSKI PARK
 NAUKOWO-TECHNOLOGICZNY

PHP
 Hypertext Preprocessor

Architektury rozwiązań informatycznych

- peer to peer
- P2P





Lekcja 1 – architektury rozwiązań informatycznych

- peer to peer - rozproszona architektura której założeniem jest możliwość połączenia się wszystkich aplikacji między sobą

P2P zakłada możliwość połączenia się wszystkich aplikacji między sobą. Każdy element architektury jest na tym samym równorzędym z innymi poziomem. Każdy element architektury jest zatem i klientem i serwerem. W przypadku odłączenia jednego elementu transmisja nadal jest prowadzona z wykorzystaniem zasobów innych elementów.

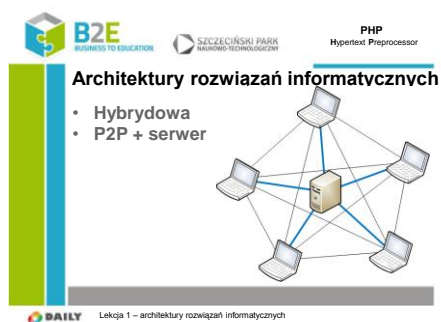
Wadą takiego rozwiązania jest konieczność przepłytywania wszystkich elementów o potrzebne

Moduł szkoleniowy SQL - Structured Query Language str. 11

Człowiek - najlepsza inwestycja

zasoby (np. pliki).

Slide 6



- hybrydowa - architektura mieszana serwer + peer to peer.

Architektura pozbawiona wady P2P, serwer pełni rolę bazy wiedzy i informacji na którym elemencie znajdują się określone zasoby. Klient nie pyta się wszystkich elementów architektury o potrzebne zasoby, zdobywa je odpytując serwer centralny. Ten w odpowiedzi kieruje klienta do właściwego hosta.

Slide 7



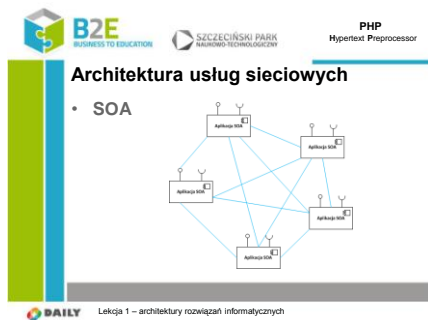
SOA jest koncepcją tworzenia systemów informatycznych w której poszczególne moduły oprogramowania (aplikacje SOA) działają autonomicznie udostępniając zestaw usług do komunikacji.

Usługi implementują funkcjonalność potrzebną klientowi lub innej aplikacji SOA. Aplikacja SOA działa na zasadzie skrzynki, przykrywając szczegóły implementacji, zapewniając ściśle określoną funkcjonalność.

Usługą może być każdy, niezależnie działający program zapewniający poprzez interfejs dostęp do określonej funkcjonalności

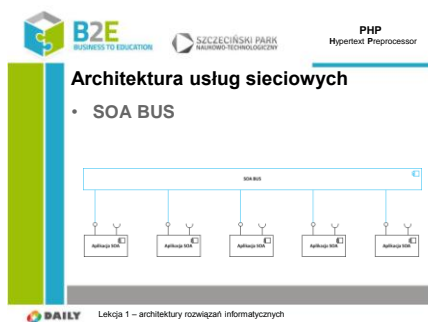


Slide 8



W szczególności, możliwe jest współdziałanie między sobą wielu aplikacji SOA (np. systemy zintegrowane). Implementacja połączenia każdy z każdym jest bardzo skomplikowana i pracochłonna...

Slide 9



...dlatego stosuje się rozwiązania typu ESB (Enterprise SOA Bus) zapewniające możliwość rejestracji usług każdej aplikacji SOA oraz definiowania procesów integracyjnych pomiędzy aplikacjami.

Slide 10



Komunikacja między aplikacjami SOA i realizacja interfejsów może zostać wykonana na wiele sposobów. Przybliżymy następujące implementacje SOA:

SOAP

WebService

REST

DCOM

COBRA

WCF

Slide 11




Implementacje SOA

SOAP

- Protokół zdalnego dostępu
- Format na bazie XML



PHP
Hypertext Preprocessor

Lekcja 1 – architektury rozwiązań informatycznych

SOAP – (Simple Object Access Protocol) protokół służący do komunikacji między systemami wykorzystujący format XML. Dokument SOAP ma określoną strukturę i składa się z:



Koperty (envelope)

Nagłówka (header)

I ciała komunikatu (body) – w którym zawarto informację jaką usługę i metodę należy wywołać

Transmisję SOAP można realizować protokołami HTTP lub RPC

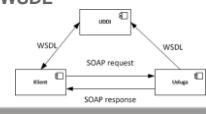
Slide 12

Implementacje SOA

WebServices

- Protokół HTTP/HTTPS
- Komunikat XML (SOAP)
- Opis interfejsu WSDL



PHP
Hypertext Preprocessor




Lekcja 1 – architektury rozwiązań informatycznych

Web service jest metodą komunikacji pomiędzy dwoma urządzeniami/aplikacjami w sieci z wykorzystaniem protokołu HTTP/HTTPS. Formatem komunikatu jest SOAP (Simple Object Access Protocol) – dokument XML o określonej strukturze

Jest wyposażona w mechanizm opisu usługi WSDL (Web Services Description Language). Pozwala on na opis usługi w taki sposób żeby aplikacja kliencka (klient) mógł poprawnie wczytać format zapytania i format odpowiedzi. Definicje poszczególnych usług i metod mogą być częścią usługi albo mogą zostać wyniesione do pośrednika (broker) UDDI (Universal

Description Discovery and Integration).

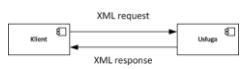
Slide 13

Implementacje SOA

REST

- Protokół HTTP/HTTPS
- Standardowe operacje HTTP: GET, PUT, POST, DELETE
- Komunikat XML






Lekcja 1 – architektury rozwiązań informatycznych

Odmianą Web service jest REST (REpresentational State Transfer). W odróżnieniu od WSDL w REST nie ma opisu formatu zapytania i odpowiedzi. Zamiast tego wykorzystywane są standardowe operacje HTTP jak GET, POST, PUT, DELETE pozwalające na realizację operacji CRUD (Create, Read, Update, Delete).

Założeniem REST jest brak stanowości zasobów, uproszczenie implementacji i zarządzania w porównaniu do WSDL.

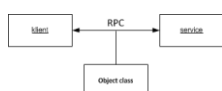
Slide 14

Implementacje SOA

DCOM,

- Serializowany komunikat
- Microsoft
- Protokół RPC



Lekcja 1 – architektury rozwiązań informatycznych



DCOM – (Distributed Component Object Model) - interfejs programistyczny zapewniający komunikację między aplikacjami na poziomie obiektów. Jest technologią firmy Microsoft przygotowaną jako konkurencję do CORBA. Technologia rozwinięta w .NET Framework stała się częścią COM+ powszechnie wykorzystywaną w Windows.

Formatem komunikatu jest serializowany obiekt określonej klasy. Zatem w celu przygotowania integracji aplikacji ze sobą wymagana jest znajomość definicja obiektów jakie używane są po stronie usługi.

Protokół: RPC, później stosowano HTTP

Problemy z implementacją, komunikacją poprzez firewallo oraz ze zwalnianiem pamięci.

Slide 15

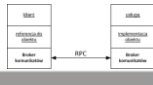



PHP
 Hypertext Preprocessor

Implementacje SOA

COBRA,

- Broker komunikatów
- Obiekt i mapowanie
- Protokół InterORB i jego odmiany



Lekcja 1 – architektury rozwiązań informatycznych

CORBA– (Common Object Request Broker Architecture) – standard zapewniający modułom oprogramowania napisanym w różnych językach, na różnych platformach do wzajemnej komunikacji. Standard został zdefiniowany przez OMG (Object Management Group).

CORBA wymaga zastosowania mechanizmu ORB (Object Request Broker) do wymiany komunikatów. Komunikaty muszą zostać prawidłowo zmapowane do wspólnego interfejsu (wprowadzony IDL – interface definition language) .

Protokół: InterORB, SSL InterORB, HTIOP – HyperText InterORB, Zipped IOP



B2E
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

DAILY
GROUP

Slide 16

Implementacje SOA

WCF,

- WSDL
- Różne formy komunikacji
- Różne formaty komunikatu

Lekcja 1 – architektury rozwiązań informatycznych

WCF – (Windows Communication Foundation) – nowoczesny zestaw narzędzi programistycznych na platformę Microsoft .NET pozwalających na budowę aplikacji SOA.

WCF mocno rozszerza paletę funkcjonalności SOA o np. dynamiczną adresację (routing), zapewnienie dostarczenia komunikatu oraz kontrolę uprawnień.

Usługa pozwala na wybór formy komunikacji np. SOAP HTTP, SOAP TCP, SOAP MQ.

Format komunikatu to uproszczony komunikat SOAP lub JSON (JavaScript Object Notation).

Slide 17

Pytania

1. Narysuj i omów architekturę klient – serwer?
2. Czy różni się architektura peer to peer od hybrydowej?
3. Co to jest usługa sieciowa?
4. Rozwiń i omów skrót SOA.
5. Wymień 4 implementacje SOA.
6. Co to jest WSDL?
7. Omów strukturę komunikatu SOAP.
8. Na jakie formy komunikacji pozwala WCF?
9. Do czego stosuje się SOA BUS?

Lekcja 1 – architektury rozwiązań informatycznych

WCF – (Windows Communication Foundation) – nowoczesny zestaw narzędzi programistycznych na platformę Microsoft .NET pozwalających na budowę aplikacji SOA.

WCF mocno rozszerza paletę funkcjonalności SOA o np. dynamiczną adresację (routing), zapewnienie dostarczenia komunikatu oraz kontrolę uprawnień.

Usługa pozwala na wybór formy komunikacji np. SOAP HTTP, SOAP TCP, SOAP MQ.

Format komunikatu to uproszczony komunikat SOAP lub JSON (JavaScript Object Notation).

Moduł szkoleniowy SQL - Structured Query Language str. 17

Człowiek - najlepsza inwestycja



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego



9.1.3 Ćwiczenia

Ćwiczenie zostało przedstawione na slajdzie 16.

9.1.4 Opis założonych osiągnięć ucznia

Uczeń powinien znać i potrafić omówić (narysować) podstawowe architektury aplikacji sieciowych oraz posiadać wiedzę na temat podanych implementacji architektury SOA.

9.2 Lekcja 2 - Protokół http (budowa protokołu HTTP 1.1, nagłówki żądania i odpowiedzi, komunikacja http)

9.2.1 Cel lekcji

Celem lekcji jest zapoznanie ucznia z protokołem HTTP, a w szczególności jego wersji 1.1. Pokazanie schematu tworzenia żądań i rozumienia odpowiedzi od serwera. Poznanie metod protokołu HTTP oraz sposobu komunikacji pomiędzy klientem i serwerem. Zastosowanie podstawowych metod \$_GET oraz \$_POST w języku PHP.

9.2.2 Treść - slajdy z opisem

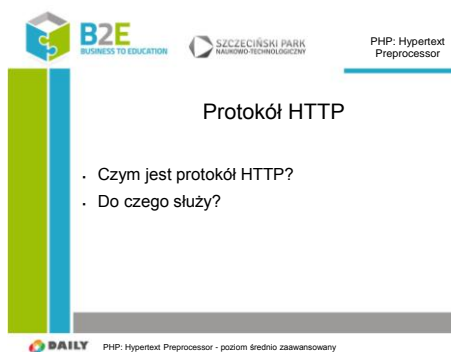
Slide 1



PHP: Hypertext Preprocessor
- poziom średnio zaawansowany
Lekcja 2
Protokół http

Słowa wstępu

Slide 2

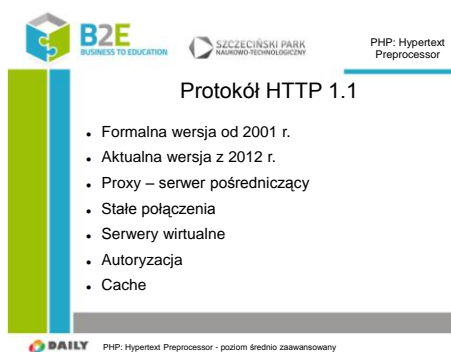


Protokół HTTP

- Czym jest protokół HTTP?
- Do czego służy?

Protokół HTTP (ang. Hypertext Transfer Protocol) – to najpopularniejszy protokół wykorzystywany w sieciach komputerowych. Mówiąc prostym językiem jest to ścisły schemat postępowania (zbiór zasad) dla urządzeń komunikacyjnych w jaki sposób przesyłać dokumenty hipertekstowe. Rozpowszechniony dzięki sieci WWW (ang. World Wide Web), za jego pomocą odbywa się przesyłanie danych oraz komunikacja w większości aplikacji internetowych.

Slide 3



Protokół HTTP 1.1

- Formalna wersja od 2001 r.
- Aktualna wersja z 2012 r.
- Proxy – serwer pośredniczący
- Stałe połączenia
- Serwery wirtualne
- Autoryzacja
- Cache

Protokół HTTP w wersji 1.1 powstał w 2001 r. Obecny standard pochodzi z grudnia 2012 r.

Obecna wersja wprowadziła szereg zmian w stosunku do poprzedniej. Możemy wykorzystać protokół jako protokół pośredniczący np. pomiędzy klientem, a innym protokołem np. SMTP albo FTP. Dzięki protokołowi TCP/IP najnowszy standard wspiera stałe połączenia z serwerem. Możliwa jest

Moduł szkoleniowy SQL - Structured Query Language str. 19

Człowiek - najlepsza inwestycja

również obsługa serwerów wirtualnych. Stworzono mechanizm autoryzacji poprzez HTTP oraz mechanizm Cache'owania czyli przechowywania poprzednich wyników.

Slide 4




PHP: Hypertext Preprocessor

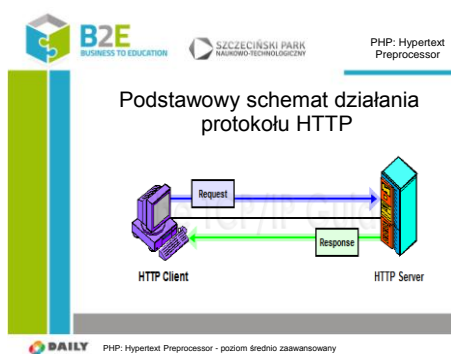
Sposób działania

- Wykorzystywany rodzaj architektury
- Bezstanowość protokołu
- Mechanizmy sesji i ciasteczek


PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Protokół HTTP wykorzystuje architekturę klient – serwer, klient(w tym przypadku przeglądarka) wysyła żądanie do serwera, który wysyła odpowiedź w zależności od otrzymanej wcześniej treści. Standaryzuję on formę zapytań oraz odpowiedzi. W związku z tym, że protokół nie zapamiętuje informacji o wcześniejszych transakcjach(transakcja = żądanie + odpowiedź), określa się go mianem protokołu bezstanowego. Brak możliwości przechowywania informacji jest swoistą przeszkodą, jeśli chodzi o komunikacje w sieci (np. autentykacja), dlatego istnieją inne mechanizmy przechowujące informacje('cookies' – po stronie klienta, 'session' po stronie serwera).

Slide 5



Slide 6

B2E **SZCZECIŃSKI PARK** **PHP: Hypertext Preprocessor**

Opis podstawowego schematu działania protokołu HTTP

1. Nawiązanie połączenia
2. Wysłanie żądania HTTP
3. Otrzymanie odpowiedzi

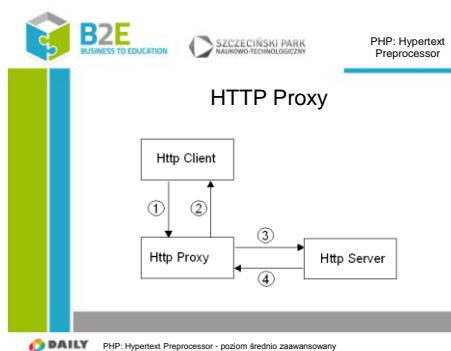
DAILY PHP: Hypertext Preprocessor - poziom średnio zaawansowany

1. Nawiązanie połączenia – w pierwszej kolejności klient używając adresu serwera(URL lub IP), nawiązuje połączenie najczęściej za pomocą protokołu TCP/IP. Domyślny port 80.

2. Następnie klient wysyła żądanie HTTP do serwera, które jest ciągiem znaków ASCII zakończona znakami „CR LF”

3. Ostatnim krokiem komunikacji jest otrzymanie odpowiedzi od serwera w postaci ciągu znaków w określonym standardzie np. HTML

Slide 7



Slide 8

B2E **SZCZECIŃSKI PARK** **PHP: Hypertext Preprocessor**

Opis działania HTTP Proxy

1. Nawiązanie połączenia z serwerem Proxy
2. Wysłanie żądania HTTP
3. Translacja adresu
4. Przekazanie żądania
5. Przekazanie odpowiedzi
6. Otrzymanie odpowiedzi



DAILY PHP: Hypertext Preprocessor - poziom średnio zaawansowany

1. Nawiązanie połączenia – w pierwszej kolejności klient używając adresu serwera(URL lub IP), nawiązuje połączenie najczęściej za pomocą protokołu TCP/IP. Domyślny port 80.
2. Następnie klient wysyła żądanie HTTP do serwera, które jest ciągiem znaków ASCII zakończona znakami „CR LF”

3. Translacja adresu zewnętrznego na adres wewnętrzny przez serwer Proxy.
4. Przekazanie żądania HTTP z serwera Proxy do wewnętrznego serwera HTTP.
5. Wysłanie odpowiedzi przez serwer wewnętrzny poprzez serwer zewnętrzny do klienta.
6. Ostatnim krokiem komunikacji jest otrzymanie odpowiedzi od serwera w postaci ciągu

znaków w określonym standardzie np. HTML

Slide 9





PHP: Hypertext Preprocessor



Przykładowe żądanie HTTP (metoda POST)

```
POST /enlighten/calais.aspx/Enlighten HTTP/1.1
Host: api.opencalais.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length

licenseID=string&content=string&paramsXML=string
```

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Slide 10





PHP: Hypertext Preprocessor

Przykładowa odpowiedź HTTP (XML)

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://clearforest.com/">string</string>
```

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Slide 11




PHP: Hypertext Preprocessor

Metody HTTP

- GET
- HEAD
- POST
- DELETE
- TRACE
- PUT
- OPTIONS

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Metody służą do komunikacji z serwerem poprzez protokół HTTP. Najpopularniejsze metody wykorzystywane w skryptach PHP to POST oraz GET. Są one dostępne w podstawowych składnikach aplikacji jak tablice superglobalne, formularze oraz adres URI. HTTP posiada jednak więcej metod, którymi możemy się w odpowiedni sposób posłużyć.



B2E
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY



Slide 12



PHP: Hypertext
Preprocessor



Metoda GET

Metoda GET służy do pobierania zasobu wskazanego przez URI, może mieć postać warunkową jeśli w nagłówku występują pola warunkowe takie jak "If-Modified-Since"

Slide 13



PHP: Hypertext
Preprocessor



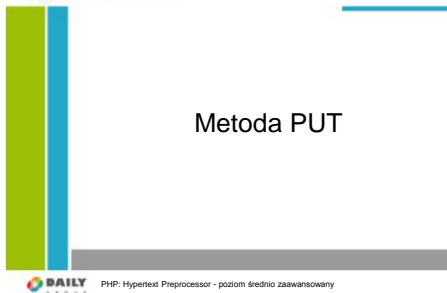
Metoda POST

Metoda ta służy do przyjmowania danych przesłanych przez klienta np. podczas przetwarzania formularzy

Slide 14



PHP: Hypertext
Preprocessor



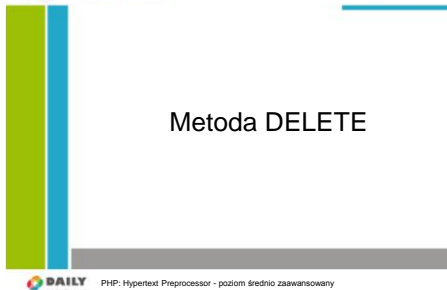
Metoda PUT

Metoda PUT służy do przesyłania danych w postaci pliku od klienta do serwera

Slide 15



PHP: Hypertext
Preprocessor



Metoda DELETE

Metoda DELETE wykorzystywana jest jako żądanie usunięcia zasobu z serwera, ze względów bezpieczeństwa poprzedzona jest autoryzacją





Slide 16



Metoda OPTIONS służy do pobierania informacji o opcjach na danym kanale komunikacyjnym dla zarejestrowanego użytkownika

Metoda OPTIONS



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Slide 17



Metoda HEAD służy do pobierania informacji o zasobie, sprawdzania jego dostępności

Metoda HEAD



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Slide 18



Metoda TRACE służy do diagnostyki, analizy kanału komunikacyjnego

Metoda TRACE



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Slide 19



Przykład zastosowania metody GET
Metoda ta przekazuje dane poprzez konkatencję parametrów z adresem URI:

np. <http://adres.pl/index.php?action=default>

Setter:

```
$action = 'default';
```

```
$url = "?action=".$action;
```

Getter:

```
$action = $_GET['action'];
```

Podstawowy mechanizm w języku PHP, zastosowania żądań typu GET oraz jednocześnie najłatwiejsza metoda skomunikowania ze sobą skryptów na serwerze HTTP. Wykorzystujemy ją dodając parametry do adresu skryptu.



PHP: Hypertext Preprocessor - poziom średnio zaawansowany





Slide 20



PHP: Hypertext
Preprocessor

Przykład zastosowania metody POST
Metoda najczęściej wykorzystywana przy przysyłaniu formularzy:

Setter:

```
<form action="adres_skryptu" method="post">
<input name="field_name" type="text" />
</form>
```

Getter:

```
$form_data = $_POST['field_name']
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Drugi podstawowy mechanizm komunikacji dostępny w języku PHP, najczęściej stosowany do przetwarzania formularzy lub asynchronicznych zapytań HTTP, czyli technologii AJAX. W przeciwieństwie do metody GET, której limit długości adresu ustalany jest przez przeglądarkę klienta. Limit długości tablicy POST ustalany jest przez konfigurację serwera. Metoda POST służy również do przysyłania plików w języku PHP.

Slide 21



PHP: Hypertext
Preprocessor

Inne metody HTTP



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Poza standardowymi metodami jak GET i POST dostępnymi w samym języku PHP. Możemy tworzyć własne zapytania HTTP korzystając przykładowo z rozszerzenia HTTP języka PHP lub zdefiniowanych przez nas obiektów, bądź funkcji. np. biblioteka cURL wspiera przysyłanie plików metodą PUT. Istnieją również architektury, które ułatwiają komunikację z serwerem, udostępniając odpowiednie API, jak architektura REST, SOAP czy Web Service.

Slide 22



PHP: Hypertext
Preprocessor

Nagłówki żądań http



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Żądanie HTTP realizuje się poprzez przesłanie odpowiednich nagłówków oraz parametrów do serwera, zakończonych znakami powrotu karetki oraz nowej linii. HTTP 1.1 pozwala na wysyłanie potokiem kilku żądań naraz. Nagłówki służą jako parametry żądania np. w jakim formacie wysyłamy żądanie i w jakim formacie odpowiedzi się spodziewamy.





Slide 23

Budowa żądania HTTP

1. Nazwa metody oraz standard np.
GET / HTTP/1.1
2. Nagłówki wymagane np.
Host: jakishost.pl
3. Nagłówki opcjonalne np.
Accept: text/xml
4. Dane np.
5. *CR + LF*

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

- W pierwszej kolejności wysyłamy nazwę metody, która w naszym przykładzie jest to metoda GET oraz nazwa standardu z jakim nasza prośba ma być przetwarzana.
- Następnie wysyłamy nagłówki wymagane przez standard w tym przypadku HOST, który jest wymagany ze względu na możliwość stworzenia Wirtualnych Hostów w obrębie jednego adresu IP.
- W dalszej części znajdują się nagłówki opcjonalne w przykładzie rodzaj akceptowalnych przez klienta danych.
- Jeżeli metoda przesyła dane np. POST umieszczamy je w tym właśnie miejscu.
- Kończymy żądanie znakiem powrotu karetki oraz znakiem przejścia do nowej linii.

Slide 24

Budowa odpowiedzi HTTP

1. Kod odpowiedzi HTTP
HTTP/1.1 200 OK
2. Nagłówki wymagane np.
Date: Thu, 20 Dec 2001 12:04:30 GMT
3. *CR + LF*
4. Zwrócone dane

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

- Pierwszym elementem jest kod odpowiedzi w tym przypadku wszystko jest prawidłowo, żądanie zostało zaakceptowane i zwrócone zostały dane.
- Nagłówki odpowiedzi w przykładzie czas serwera.
- Zakończenie nagłówków odpowiedzi.
- W tym miejscu przekazane są dane zwrócone z serwera HTTP.

Slide 25

Dziękujemy za uwagę

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

9.2.3 Ćwiczenia

Jako ćwiczenia należy zlecić uczniu napisanie przykładów użycia metod HTTP znajdujących się na slajdach.

9.2.4 Opis założonych osiągnięć ucznia

Uczeń powinien poznać i umieć opowiedzieć o metodach protokołu HTTP, potrafił wyjaśnić na czym polega komunikacja pomiędzy klientem, a serwerem. Powinien umieć opowiedzieć z czego składa się żądanie oraz odpowiedź HTTP.

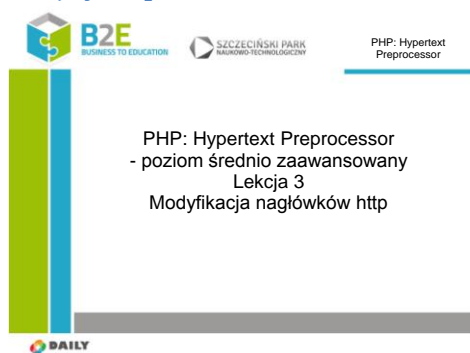
9.3 Lekcja 3 - Modyfikacja nagłówków http (Lista nagłówków, Wykorzystanie funkcji header, Zastosowania w skryptach PHP)

9.3.1 Cel lekcji

Celem lekcji jest nauczenie uczestników nagłówków http. Powinni potrafić używać ich w języku php przy użyciu funkcji header.

9.3.2 Treść - slajdy z opisem

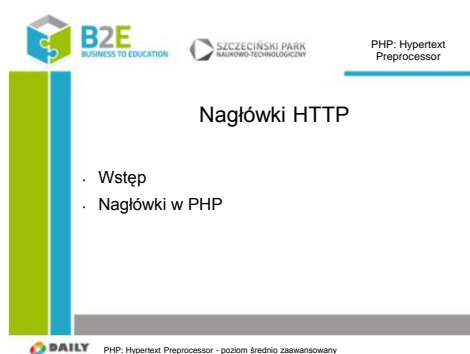
Slide 1



Slide 1 content: The slide features a green and blue vertical bar on the left with the B2E logo. The main text reads: 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany', 'Lekcja 3', and 'Modyfikacja nagłówków http'. Logos for B2E, Szczeciński Park Naukowo-Technologiczny, and Daily Group are present at the top and bottom.

Słowa wstępu

Slide 2



Slide 2 content: The slide features a green and blue vertical bar on the left with the B2E logo. The main text reads: 'Nagłówki HTTP', followed by a bulleted list: '· Wstęp' and '· Nagłówki w PHP'. Logos for B2E, Szczeciński Park Naukowo-Technologiczny, and Daily Group are present at the top and bottom.

Nagłówki HTTP w języku PHP, wykorzystywane są do komunikacji pomiędzy serwerem, a stroną WWW. Do ich wysyłania najczęściej stosujemy funkcję 'header'. Używać ich może w różnych celach np. do autoryzacji HTTP, odświeżenia strony, bądź przekierwania przez serwer na inną stronę.



Slide 3



Dostępne nagłówki


- Standardowe
- Niestandardowe

PHP: Hypertext Preprocessor

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Nagłówki HTTP w języku PHP, dzielimy na standardowe i niestandardowe. Standardowe to te, które znajdują się w specyfikacji HTTP w wersji 1.1, a niestandardowe to takie nagłówki, które są dostępne, lecz nie znajdują się w specyfikacji.

Slide 4



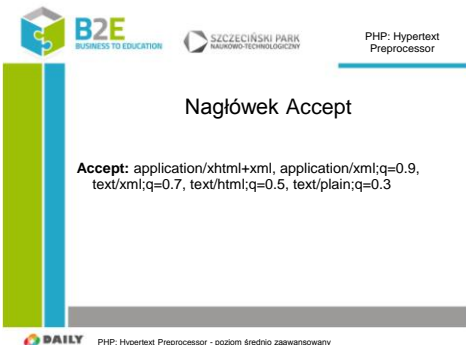
Lista nagłówków

PHP: Hypertext Preprocessor

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Nagłówki HTTP w języku PHP, wykorzystywane są do komunikacji pomiędzy serwerem, a stroną WWW. Do ich wysyłania najczęściej stosujemy funkcję 'header'. Używać ich może w różnych celach np. do autoryzacji HTTP, odświeżenia strony, bądź przekierwania przez serwer na inną stronę.

Slide 5



Nagłówek Accept

Accept: application/xhtml+xml, application/xml;q=0.9, text/xml;q=0.7, text/html;q=0.5, text/plain;q=0.3

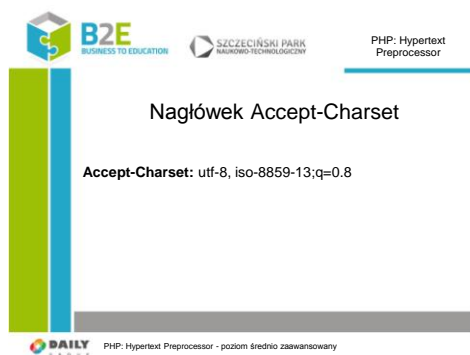
PHP: Hypertext Preprocessor

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Służy do określenia listy akceptowalnych przez przeglądarkę typów MIME dokumentu, oraz opcjonalnie hierarchii każdego typu. Liczby podane po ;q= powinny mieć wartości od 0 do 1, co 0.1. Jako separator miejsc dziesiętnych użyta musi być kropka. W przypadku braku zdefiniowanej wartości ;q= przyjmowana jest wartość: ;q=1 (równoznaczna z ;q=1.0). Powyższy przykład ustala najwyższy priorytet typowi application/xhtml+xml. Jeżeli serwer nie posiada dokumentu w formacie application/xhtml+xml, to prześle plik w formacie application/xml – ogólnym dla podjęzyków XML, takich jak XHTML, SVG. Jeżeli na serwerze nie

ma również dokumentu w tym formacie, dane zostaną przesłane jako text/xml – czyli ogólne dane w formacie XML, w ostateczności, jeżeli żadnego z tych typów nie posiada, serwer wyśle dane w formacie text/plain, czyli zwykłego tekstu. Typy MIME mogą być określone za pomocą symboli wieloznacznych np.: text/*, */*.


Slide 6



PHP: Hypertext Preprocessor

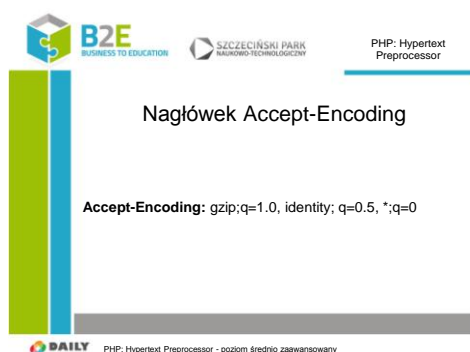
Nagłówek Accept-Charset

Accept-Charset: utf-8, iso-8859-13;q=0.8

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Określa preferowane przez przeglądarkę formaty kodowania. Wartości ;q= interpretowane są na takich samych zasadach, jak w przypadku typów MIME z nagłówka Accept.


Slide 7



PHP: Hypertext Preprocessor

Nagłówek Accept-Encoding

Accept-Encoding: gzip;q=1.0, identity;q=0.5, */q=0

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Ten nagłówek określa kodowanie, za pomocą którego zostanie przesłana zawartość. Stosowane zwykle do kompresji treści.



B2E
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY



Slide 8

Slide 8 content: Nagłówek Accept-Language

Accept-Language: pl, en-us;q=0.7

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Określa w jakim języku użytkownik przeglądarki życzy sobie czytać strony

Slide 9

Slide 9 content: Nagłówek Accept-Ranges

Accept-Ranges: bytes lub none

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Ten nagłówek jest ustawiany przez przeglądarki i programy ułatwiające ściąganie plików. Określa czy klient potrafi (wartość bytes) lub nie (wartość none) odczytywać pliki przesyłane w częściach.

Slide 10

Slide 10 content: Nagłówek Allow

Allow: GET, POST, HEAD

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Lista metod na które serwer ma zezwolić.



Slide 11

Nagłówek Authorization

Authorization: Basic QWxhZGRpbjpvGVuIHNlc2FZQ==

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Przesyła nagłówek do autoryzacji, najczęściej zakodowane login i hasło.

Slide 12

Nagłówek Cache-Control

Cache-Control: no-cache/no-store/max-age=3600/min-fresh=3600

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Określa regułę według której przeglądarka może (lub nie może) przechowywać stronę w pamięci podręcznej.

Dostępne wartości to:

- no-store – strona nie może być przechowywana.
- no-cache – strona może być przechowywana pod warunkiem każdorazowego sprawdzania jej aktualności.

Te dwa nagłówki mają szczególne znaczenie dla serwerów proxy. Pierwszy nakazuje przesyłać dokument bezwzględnie, natomiast drugi pozwala na skorzystanie z kopii w pamięci podręcznej, jeżeli dokument źródłowy się nie zmienił.

- max-age=liczba_sekund – oznacza, że można skorzystać z cache pod warunkiem iż nie jest przechowywane dłużej jak liczba_sekund. Jeżeli jest starsze przeglądarka powinna ściągnąć nowszą wersję.
- min-fresh=liczba_sekund – nakazuje przeglądarce ściągnąć dokument jeżeli jego wiek nie przekracza liczby_sekund + aktualna_data



B2E
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

DAILY
GROUP

Slide 13

Slide 13 content: Nagłówek Connection

Connection: keep-alive/close

PHP: Hypertext Preprocessor

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Nagłówek określa, czy zamknąć połączenie po zapytaniu, czy je utrzymywać.

Slide 14

Slide 14 content: Nagłówek Cookie

Cookie: ciastko1=wartosc1; ciastko2=wartosc2

PHP: Hypertext Preprocessor

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

W tym nagłówku przesyłane są wszystkie ciasteczka ustawione w przeglądarce.

Slide 15

Slide 15 content: Nagłówek Content-Encoding

Content-Encoding: gzip

PHP: Hypertext Preprocessor

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Określa typ kodowania pliku.



B2E
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

DAILY
GROUP

Slide 16

Slide 16 content: Nagłówek Content-Language

Content-Language: en, pl

PHP: Hypertext Preprocessor

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Język dokumentu przesłanego przez serwer. Może być określony więcej niż jeden język. Wtedy język główny strony jest określony jako ostatni na tej liście.

Slide 17

Slide 17 content: Nagłówek Content-Length

Content-Length: 1000

PHP: Hypertext Preprocessor

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Określa rozmiar zawartości w bajtach

Slide 18

Slide 18 content: Nagłówek Content-Range

Content-Range: bytes 123-456/5000

PHP: Hypertext Preprocessor

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Określa zakres przesłanej części pliku w tym przypadku bajty od 123 do 456 z 5000(cały plik).

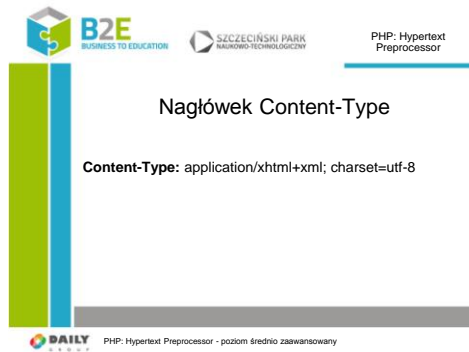


B2E
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

DAILY
GROUP

Slide 19



Tym nagłówkiem serwer informuje przeglądarkę, w jakim formacie i stronie kodowej wysyłany jest dokument.

Slide 20



Data serwera.

Slide 21



Data powyżej której dokument będzie nieaktualny.



Slide 22



Nagłówek określający z dla jakiej domeny określony został URI (do serwera bowiem nie dociera nazwa domenowa, ale jedynie adres IP). Nagłówek wymagany w HTTP 1.1. Dzięki niemu można uruchomić kilka serwerów wirtualnych na jednym adresie IP.

Slide 23



Nagłówek nakazuje serwerowi przesłać dokument tylko jeżeli został zmodyfikowany od danej daty. Jeżeli dokument się nie zmienił serwer powinien wysłać przeglądarce kod odpowiedzi 304.

Slide 24



Nagłówek serwera informujący o ostatniej aktualizacji dokumentu.



B2E
BUSINESS TO EDUCATION



SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY



Slide 25

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

Nagłówek Location

Location: http://serwer/

DAILY GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Wymusza przekierowanie na określony adres. Dla kodu odpowiedzi 201 adres oznacza miejsce nowo utworzonego zasobu. Dla kodów 3xx podany adres oznacza przekierowanie stałe. Od tej pory przeglądarka powinna korzystać z nowego adresu.

Slide 26

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

Nagłówek Range

Range: bytes=500-999

DAILY GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Określa jaką część pliku oczekuje przeglądarka. Podawany w formacie bytes=pozycja_początkowa-pozycja_końcowa

Slide 27

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

Nagłówek Referer

Referer: http://serwer/katalog/plik.html

DAILY GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany



B2E
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

DAILY
GROUP

Slide 28

Slide 28 content: Nagłówek Refresh

Refresh: czas; url=adres

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Ustawia automatyczne przekierowanie w przeglądarce na podany adres po określonym czasie

Slide 29

Slide 29 content: Nagłówek Retry-After

Retry-After: Fri, 31 Dec 1999 23:59:59 GMT

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Używany razem z kodem odpowiedzi 503 (Serwer zajęty). Określa czas po którym serwer będzie w stanie odpowiedzieć. Czas może być podany w formie konkretnej daty (jak w przykładzie) lub liczby sekund.

Slide 30

Slide 30 content: Nagłówek Server

Server: Apache/2.0.50

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Nagłówek identyfikujący serwer i użyte w nim oprogramowanie.



Slide 31

Nagłówek Set-Cookie

Set-Cookie: ciastko1=wartosc1; ciastko2=wartosc2

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Nagłówek wysłany
od serwera nakazuje przeglądarce
ustawienie określonych ciasteczek.

Slide 32

Nagłówek Transfer-Encoding

Transfer-Encoding: chunked

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Określa w jaki sposób serwer przesłał
zawartość dokumentu.

Dostępne metody to:

- chunked

Oznacza przesyłanie zawartości
kawałkami. Każdy z nich powinien
mieć postać: wielkość treść wielkość –
jest to długość treści w bajtach
zapisane w formie szesnastkowej.
Przykład: 1a
abcdefghijklmnpqrstuvwxy 10
1234567890abcdef 0 Zero w ostatniej
linijce określa zerowej długości
kawałek i kończy transmisję.

- identity

Oznacza tekst nieskompresowany

- gzip

Oznacza dane skompresowane
algorytmem ZIP

- compress

Dane skompresowane algorytmem
LZW



B2E
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

DAILY
GROUP

Slide 33

B2E **SZCZECIŃSKI PARK** **DAILY**
BUSINESS TO EDUCATION NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

Nagłówek User-Agent

User-Agent: Mozilla/5.0 (X11; U; Linux i686; pl; rv:1.8.0.1) Gecko/20060124 Firefox/1.5.0.1

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Identyfikuje przeglądarkę, silnik html oraz System operacyjny.

Slide 34

B2E **SZCZECIŃSKI PARK** **DAILY**
BUSINESS TO EDUCATION NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

Nagłówek WWW-Authenticate

WWW-Authenticate: rodzaj_autentyfikacji

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Przesyłany razem z kodem 401. Określa sposób w jaki ma zostać przeprowadzona identyfikacja użytkownika.

Slide 35

B2E **SZCZECIŃSKI PARK** **DAILY**
BUSINESS TO EDUCATION NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

Funkcja getallheaders


- array **getallheaders** (void)
- array **apache_response_headers** (void)

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Funkcja 'getallheaders' pobiera wszystkie nagłówki żądania http i parsuje je do tablicy PHP, a funkcja 'apache_response_headers', parsuje wszystkie nagłówki odpowiedzi http.




Slide 36

PHP: Hypertext Preprocessor

Funkcja header



```
void header ( string $string [, bool $replace = true [, int $http_response_code ]] )
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Funkcja 'header' służy do przesyłania nagłówka http dokumentu. Pierwszym argumentem jest treść nagłówka w postaci ciągu znaków PHP. Drugim argumentem jest argumentem opcjonalnym, mówi o tym, czy funkcja ma zastąpić nagłówki tego samego typu przygotowany przez serwer. Kolejnym argumentem opcjonalnym jest narzuca serwerowi kod odpowiedzi.


Slide 37

PHP: Hypertext Preprocessor

Przykłady zastosowania header()



```
<?php
header("HTTP/1.0 404 Not Found");
?>
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Przy odpowiedniej konfiguracji serwera HTTP, język PHP może zwracać komunikaty o braku zasobów np. strony o podanym adresie. W tym wypadku znany wszystkim błąd 404.


Slide 38

PHP: Hypertext Preprocessor

Przykłady zastosowania header()

```
<?php
header("Location: http://przyklad.pl/");
?>
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Proste przekierowanie HTTP na podany adres.



Slide 39

Przykłady zastosowania header()

```
<?php
$file='przyklad.php';
header('Content-disposition: attachment;
filename='.basename($file));
header('Content-type: text/plain');
readfile($file);
?>
```

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Przykład dotyczy wykorzystania nagłówków do pobrania pliku typu PHP jako tekst.

Slide 40

Przykłady zastosowania header()

```
header('Cache-Control: no-cache, no-store, max-age=0,
must-revalidate');
header('Expires: Mon, 26 Jul 1997 05:00:00 GMT');
header('Pragma: no-cache');
```

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Powyższy przykład spowoduje wyłączenie cache'u.

Slide 41




Przykłady zastosowania header()

```
<?php
header('Content-Type: text/html; charset=utf-8');
echo 'łączenie';
?>
```

PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Powyższy przykład ustawia typ kodowanie pliku na utf-8.

Slide 42



Przykład zastosowania nagłówka HTTP strony HTML w PHP

```
<?php  
echo "<meta http-equiv='refresh'  
content='0';url=http://przyklad.pl'>";  
?>
```

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Przekierowanie lub odświeżenie strony z poziomu nagłówków dokumentu html.

Slide 43



Dziękujemy za uwagę

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

9.3.3 Ćwiczenia

Jako ćwiczenia należy przećwiczyć z uczniami przykłady podane na slajdach.

9.3.4 Opis założonych osiągnięć ucznia

Po tej lekcji uczniowi będą znali nagłówki protokołu HTTP oraz będą potrafili stosować je w swoich skryptach.

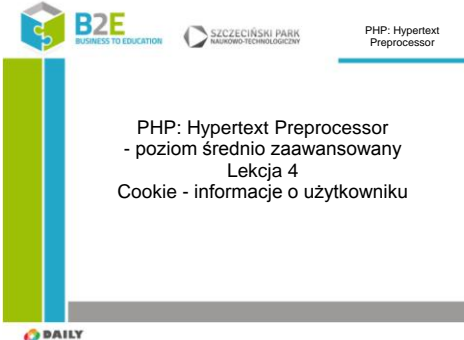
9.4 Lekcja 4 – Cookie - informacje o użytkowniku (Budowa i działanie cookie, Używanie cookie w aplikacjach, Definiowanie i obsługa z poziomu PHP, Bezpieczeństwo)

9.4.1 Cel lekcji

Celem lekcji jest wyjaśnienie uczniom na czym polega mechanizm ciasteczek i jak stosować go w języku PHP. Opisano również wady i zalety stosowania tego mechanizmu oraz do czego należy go stosować.

9.4.2 Treść - slajdy z opisem

Slide 1

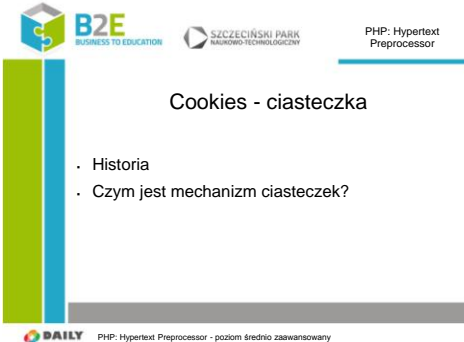


PHP: Hypertext Preprocessor

PHP: Hypertext Preprocessor
- poziom średnio zaawansowany
Lekcja 4
Cookie - informacje o użytkowniku

Słowa wstępu

Slide 2



Cookies - ciasteczka

- Historia
- Czym jest mechanizm ciasteczek?

Mechanizm ciasteczek został wymyślony przez pracownika firmy Netscape Communications.

Ciasteczko to niewielka porcja danych wysyłana przez serwer WWW i zapisywana w danych przeglądarki po stronie klienta. Dostęp do nich ma tylko serwer, który je wysłał, czyli mówiąc prosto każda domena ma swoje własne ciasteczka po stronie klienta, o ile mechanizm jest włączony i ciasteczka dla konkretnego adresu są akceptowane. Obecnie wykorzystuje się je głównie do prowadzenia statystyk, bądź zliczania wyświetleń reklam.



Slide 3

Cookies w PHP

- Gdzie można je znaleźć – tablica `$_COOKIE`?
- Jak z nich korzystać?

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Wszystkie ciasteczka dla danej domeny przechowywane są w superglobalnej tablicy `$_COOKIE`. PHP udostępnia kilka funkcji pomocniczych do mechanizmu ciasteczek np. `setcookie`.

Slide 4

Set Cookie – Tworzymy ciasteczko



- `bool setcookie (string $name [, string $value [, int $expire = 0 [, string $path [, string $domain [, bool $secure = false [, bool $httponly = false]]]]]])`
- `bool setrawcookie (string $name [, string $value [, int $expire = 0 [, string $path [, string $domain [, bool $secure = false [, bool $httponly = false]]]]]])`

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Ciasteczko ustawiamy(tworzymy) używając funkcji `setcookie` lub analogicznej funkcji `setrawcookie`, które różnią się tylko tym, że `setrawcookie` nie używa automatycznego enkodowania adresu url. Wymagany jest pierwszy parametr, czyli nazwa ciasteczka. Pozostałe parametry są opcjonalne, dlatego umieszczone w nawiasach kwadratowych. Drugim parametrem jest wartość ciasteczka. Kolejnym data w do której ciasteczko jest ważne w formacie Unixowym, czyli daty i czasu skonwertowanego do liczby całkowitej. Czwartym parametrem jest ścieżka pod którą ciasteczka mają być dostępne np. `/`, pozwala na dostęp w dowolnej ścieżce. Piątym parametrem jest nazwa domeny, która ogranicza dostęp do ciasteczka dla domen zewnętrznych, najbardziej zewnętrzną domeną, która pozwala na dostęp dla całej witryny jest np. `nazwa_domeny.pl`. Wszystkie domeny wewnętrzne np. `www.nazwa_domeny.pl`, będą miały dostęp do ciasteczek. Następne parametry określają dostęp do ciasteczka ze względu na używany protokół. `Secure` mówi, że ciasteczko będzie tylko dostępne jeżeli ustawione jest połączenie `HTTPS`, a `httponly`, że

jest dostępne tylko za pomocą protokołu HTTP. Funkcja zwraca true(prawdę), jeżeli ciasteczko zostało ustawione, false(falsz) w przeciwnym wypadku.

Slide 5

 **B2E**
BUSINESS TO EDUCATION  **SZCZECIŃSKI PARK**
NAUKOWO-TECHNOLOGICZNY PHP: Hypertext
Preprocessor

Read Cookie – odczytujemy ciasteczko

- Do odczytu używamy tablicy oraz klucza(nazwy)



```
if(isset($_COOKIE['cookie_name']))  
$cookie = $_COOKIE['cookie_name'];
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Odczyt ciasteczek jest o wiele prostszy, niż ich tworzenie. Posługujemy się tablicą `$_COOKIES` oraz nazwą ciasteczka. Z racji możliwości kasowania ciasteczek przez użytkowników, należy się upewnić, że owe ciasteczko istnieje, możemy wykorzystać do tego funkcję `isset`.

Slide 6

 **B2E**
BUSINESS TO EDUCATION  **SZCZECIŃSKI PARK**
NAUKOWO-TECHNOLOGICZNY PHP: Hypertext
Preprocessor

Ćwiczenie – usuwamy ciasteczko

- Wykorzystując funkcję `setcookie` ustaw ciasteczko o nazwie 'name', zawierające wartość Twojego imienia oraz następnie sprawdź czy istnieje jeżeli tak, wyświetl komunikat: Hello, 'ciasteczko_z_imieniem!' (skrypt nazwij `name.php`) Jeżeli nie ma ciasteczka lub jest puste, wypisz tylko 'Hello'. Odśwież stronę dwa razy.





PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Wykonać ćwiczenie, na następnej stronie znajduje się rozwiązanie ćwiczenia, nie pokazywać dopóki studenci nie skończą.




Slide 7

  PHP: Hypertext Preprocessor



Rozwiązanie i opis

```
<?php
if(!isset($_COOKIE['name']))
    setcookie('name', 'Stefan');
if(isset($_COOKIE['name']) and !empty($_COOKIE['name']))
    echo 'Hello, ' . $_COOKIE['name'] . '!';
else
    echo 'Hello!';
?>
```

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Rozwiązanie poprzedniego ćwiczenia. W pierwszej kolejności sprawdzamy, czy ciasteczko istnieje, jeżeli nie to je tworzymy. Następnie jeżeli ciastko istnieje i nie jest puste wypisujemy 'Hello, imię', a jeżeli istnieje i jest puste wypisujemy tylko 'Hello'.

Slide 8

  PHP: Hypertext Preprocessor



Overwrite Cookie – nadpisywanie ciasteczka

- Ponowne wywołanie funkcji setcookie, bez sprawdzenia, czy ciasteczko już istnieje

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Sprawdzenie istnienie ciasteczka za pomocą funkcji 'isset', jest rzeczą istotną, ponieważ jeżeli ustawimy ciasteczko o nazwie istniejącego ciasteczka, zostanie ono nadpisane nowym. Mechanizm ten jest wykorzystywany z tym, że dobrze rozumieć jak to działa, ze względu na możliwe problemy w aplikacjach.

Slide 9

  PHP: Hypertext Preprocessor

Delete Cookie – usuwanie ciasteczka



- `setcookie("hello", "", time()-3600);`

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Do usuwania ciasteczek najczęściej stosuje się mechanizm nadpisywania, ponieważ ciasteczka, których okres ważności się skończył, są usuwane. W powyższym przykładzie ustawiliśmy ważność ciasteczka na czas obecny minus 3600 sekund, czyli na godzinę wstecz.




Slide 10

  PHP: Hypertext Preprocessor

Ćwiczenie

- Wykorzystując skrypt napisany w poprzednim ćwiczeniu, usunąć ciasteczko i zmodyfikować skrypt w taki sposób, aby nie ustawiać ponownie ciasteczka. Odświeżyć przeglądarkę dwa razy.

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Wykonać ćwiczenie, na następnej stronie znajduje się rozwiązanie ćwiczenia, nie pokazywać dopóki studenci nie skończą.

Slide 11

  PHP: Hypertext Preprocessor



Rozwiązanie

```
<?php
if(isset($_COOKIE['name']))
    setcookie('name', "", time() - 3600);
if(isset($_COOKIE['name']) and !empty($_COOKIE['name']))
    echo 'Hello, ' . $_COOKIE['name'] . '!';
else
    echo 'Hello!';
?>
```

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Rozwiązanie poprzedniego ćwiczenia. W pierwszej kolejności sprawdzamy, czy ciasteczko istnieje, aby następnie ustawić jego datę ważności na godzinę wstecz. Następnie jeżeli istnieje, wypisujemy 'Hello, ciastko', bądź jeżeli je usunęliśmy tylko 'Hello'.

Slide 12

  PHP: Hypertext Preprocessor

Dlaczego odświeżamy przeglądarkę dwa razy?

- Ustawiamy ciastko
- Pobieramy ciastko

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Skrypty PHP wykonywane są po stronie serwera w wyniku zgłoszenia żądanie HTTP w którym wysyłamy ciasteczka dla danego serwera. Przy pierwszym żądaniu serwer otrzymuje stan ciasteczek z poprzedniego żądania, następnie wykonuje skrypt w którym ustawia ciastko. Wysyła odpowiedź HTTP z nowo ustawionym ciastkiem. Dopiero w drugim żądaniu otrzymuje ciastko ustawione w pierwszym żądaniu i może je przetworzyć.



Slide 13

Ćwiczenie – przykładowe zastosowanie

- Wykorzystując wiedzę z poprzednich slajdów i ćwiczeń. Napisać licznik odwiedzin, wykorzystujący mechanizm ciasteczek.

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Wykonać ćwiczenie, na następnej stronie znajduje się rozwiązanie ćwiczenia, nie pokazywać dopóki studenci nie skończą.

Slide 14

Rozwiązanie

```
<?php
if(!isset($_COOKIE['counter']))
    setcookie('counter', 2);
if(isset($_COOKIE['counter']))(
    setcookie('counter', $_COOKIE['counter'] + 1);
    echo 'Odwiedzasz stronę ' . $_COOKIE['counter'] . ' raz!';
) else {
    echo 'Odwiedzasz stronę 1 raz!';
}
?>
```

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Rozwiązanie poprzedniego ćwiczenia. Jeżeli ciastko licznika nie istnieje ustawiamy je na dwa, ponieważ, zostanie wyświetlone dopiero przy następnej wizycie i wyświetlamy ciąg znaków z wyrażenia warunkowe po 'else'. W przypadku gdy ciastku jest już ustawione, wyświetlamy jego wartość i zwiększamy o jeden. Kolejność nie ma znaczenia, gdyż jak wcześniej wspomniano ustawione zostanie ono dopiero w odpowiedzi HTTP po stronie klienta, aby zostać wysłane przy kolejnym żądaniu HTTP.

Slide 15

Zalety ciasteczek

- Prywatność - są dostępne tylko dla użytkownika w jego przeglądarce i nikt z zewnątrz nie ma do nich dostępu
- Użytkownik ma na nie bezpośredni wpływ, co może być również wadą, ale w niektórych przypadkach jest zdecydowanie zaletą np. możliwość ich blokowania dla konkretnych stron

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Przedyskutować wypunktowane kwestie.





B2E
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

DAILY
GROUP


Slide 16

PHP: Hypertext Preprocessor

Zalety ciasteczek



- Łatwość w ich obsłudze z poziomu języka PHP
- Ułatwienia dla użytkownika, np. nie musi wypełniać formularza, kolejny raz
- Możliwość tworzenia efektywnych kampanii reklamowych oraz prowadzenia statystyk (zaleta dla właściciela portalu)

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Przedyskutować kwestie.

wypunktowane


Slide 17

PHP: Hypertext Preprocessor

Wady ciasteczek



- Ciasteczka muszą być dołożone do każdego żądania HTTP
- Mają ograniczony rozmiar z góry przez przeglądarkę
- Bez użycia HTTPS są widoczne i niezaszyfrowane w przeglądarce
- Użytkownik może zmieniać dane w nich zawarte

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Przedyskutować kwestie.

wypunktowane


Slide 18

PHP: Hypertext Preprocessor

Wady ciasteczek




- Użytkownik może je usunąć
- Użytkownik może zablokować ciasteczka
- Zależą od przeglądarki, czyli ich bezpieczeństwo zależy od konkretnej implementacji, czasami wadliwej, wykazano luki w niektórych przeglądarkach

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Przedyskutować kwestie.

wypunktowane

Slide 19

  
Podsumowanie

- Czy warto je stosować?
- Do czego warto je stosować?

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Ciasteczka ułatwiają życie zarówno użytkownikowi, programiście oraz właścicielowi portalu. W niektórych kwestiach warto je stosować np. statystykach, autouzupełnianiu formularzy, kampaniach reklamowych czy jeżeli używamy szyfrowania np. SSL również do autoryzacji.

Slide 20

  
Dziękujemy za uwagę PHP: Hypertext Preprocessor - poziom średnio zaawansowany

9.4.3 Ćwiczenia

Ćwiczenia zostały przedstawione na slajdach. Podczas prowadzenia zajęć należy wykonywać je razem z uczniami.

9.4.4 Opis założonych osiągnięć ucznia

Uczeń powinien posiadać wiedzę na temat mechanizmu ciasteczek, poznać schemat jego działania oraz potrafić korzystać z niego w swoich skryptach.

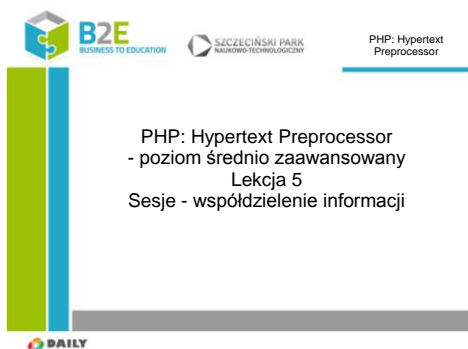
9.5 Lekcja 5 - Sesje - współdzielenie informacji (Struktura i sposób działania, Kontrola sesji w PHP, Wykorzystanie sesji w aplikacjach internetowych)

9.5.1 Cel lekcji

Celem lekcji jest zaprezentowanie mechanizmu sesji, opis jego wykorzystania w skryptach i nauczenie uczestników kursu wykorzystywać powyższy mechanizm w ich skryptach.

9.5.2 Treść - slajdy z opisem

Slide 1

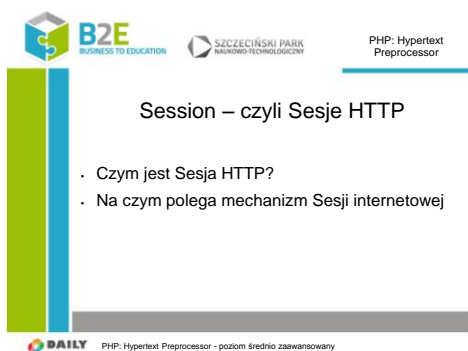


PHP: Hypertext Preprocessor

PHP: Hypertext Preprocessor
- poziom średnio zaawansowany
Lekcja 5
Sesje - współdzielenie informacji

Słowa wstępu

Slide 2





Session – czyli Sesje HTTP

- Czym jest Sesja HTTP?
- Na czym polega mechanizm Sesji internetowej

Sesja HTTP to mechanizm przechowywania danych po stronie serwera. Parametry sesji takie jak np. czas trwania sesji, znajdują się w pliku konfiguracyjnym serwera. W przeciwieństwie do mechanizmu ciasteczek znanego z poprzedniej lekcji. Sesje są niezależne od klienta, dzięki czemu dają programiście większą kontrolę nad danymi w nich przechowywanymi. Każda sesja posiada swój identyfikator, który jest unikatowy i jeden dla każdego klienta. Klientem w rozumieniu sesji jest np. przeglądarka, a nie adres komputera osoby z niej korzystającej. Konsekwencją tego jest tworzenie kilku sesji, dla jednego adresu komputera, w przypadku np. zalogowania się z różnych przeglądarek, czyli możliwość wielokrotnego logowania się na np. jedno konto, w przypadku nie zaimplementowanej możliwości jednokrotnego dostępu z poziomu jednego konta.




Slide 3

PHP: Hypertext Preprocessor

Sesje w PHP

- Superglobalna tablica \$_SESSION
- Otwieranie i zamykanie sesji



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Sesje w przeciwieństwie do ciasteczek, są mechanizmem bardziej rozbudowanym. Podobnie jak ciasteczka posiadają tablicę superglobalną, dostępną w każdym miejscu aplikacji PHP. Dane z sesji są dostępne w skrypcie od razu po ich stworzeniu(ustawieniu), nie ma potrzeby odświeżania strony, żeby korzystać z sesji trzeba ją jednak najpierw otworzyć lub stworzyć, możemy ją również zamknąć wedle naszej potrzeby.

Slide 4




PHP: Hypertext Preprocessor

Inicjalizacja sesji



- bool session_start()
- session_status()



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Aby, zainicjalizować sesję, używamy funkcji 'session_start', zwraca ona zawsze wartość true i nie przyjmuje żadnych argumentów. Aby pobrać stan sesji stosujemy funkcję 'session_status', zwraca 0 jeżeli sesje są wyłączone po stronie serwera, 1 jeżeli sesja nie istnieje lub jest nieaktywna bądź, 2 jeżeli istnieje.


Slide 5

PHP: Hypertext Preprocessor

Identyfikator sesji

- string session_id([string \$id])
- bool session_regenerate_id ([bool \$usuniecie_starej_sesji = false])



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Funkcja session_id(), zwraca nam identyfikator obecnej sesji lub jeżeli zostanie identyfikator jako parametr, ustawia identyfikator sesji, w tym przypadku należy ją wywołać przed funkcją session_start. Kolejna funkcja zastępuje obecny identyfikator sesji, nowym, a argument decyduje o tym, czy usunąć zmienianą sesję, czy nie.



Slide 6



PHP: Hypertext Preprocessor

Zamykanie sesji



- bool session_start()
- session_status()



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Aby, zainicjalizować sesję, używamy funkcji 'session_start', zwraca ona zawsze wartość true i nie przyjmuje żadnych argumentów. Aby pobrać stan sesji stosujemy funkcję 'session_status', zwraca 0 jeżeli sesja są wyłączone po stronie serwera, 1 jeżeli sesja nie istnieje lub jest nieaktywna bądź, 2 jeżeli istnieje.


Slide 7



PHP: Hypertext Preprocessor

Ćwiczenie – tworzenie sesji

- Napisz skrypt, który stworzy nową sesję oraz wyświetli jej identyfikator, którym będzie Twoje imię(nie stosuj polskich znaków).



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Wykonać ćwiczenie, na następnej stronie znajduje się rozwiązanie ćwiczenia, nie pokazywać dopóki studenci nie skończą.

Slide 8



PHP: Hypertext Preprocessor

Rozwiązanie

```
<?php
session_id('Filip');
if(session_id() == "")
    session_start();
echo session_id();
?>
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

W pierwszej kolejności ustalamy nazwę sesji na nasze imię. Następnie sprawdzamy czy sesja istnieje, jeżeli nie istnieje tworzymy nową lub startujemy wcześniej utworzoną. Na koniec wypisujemy identyfikator sesji.



Slide 9



PHP: Hypertext Preprocessor

Dodawanie danych do sesji

- session_register (stara metoda)
- \$_SESSION['key'] = \$value;



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Aby dodać dane do wcześniej rozpoczętej sesji, należy użyć drugiej metody ze slajdu. Pierwsza metoda jest przestarzała i niepoprawna w najnowszej implementacji PHP, działa w wersji PHP < 5.4. Wartości zapisujemy operując bezpośrednio na superglobalnej tablicy sesji.

Slide 10



PHP: Hypertext Preprocessor

Nadpisywanie danych w sesji


- \$_SESSION['key'] = \$value;



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Jeżeli klucz w sesji istnieje, a my dodamy od niego nowe dane, zostanie on nadpisany. Tablica sesji działa dokładnie tak jak inne tablice w języku PHP.


Slide 11



PHP: Hypertext Preprocessor

Usuwanie bieżących danych sesji

- session_destroy();



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Funkcja session_destroy usuwa tylko bieżące dane skojarzone z sesją. Nie usuwa jednak sesji, której identyfikator przechowywany jest w ciasteczku sesji.



Slide 12




PHP: Hypertext Preprocessor

Nazwa sesji



- string **session_name** ([string \$nazwa])



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Nazwa sesji jest obok identyfikatora, jest używana do identyfikacji sesji. Funkcja `session_name` podobnie jak funkcja `session_id`, służy albo do zwracania nazwy sesji albo do zmieniania jej nazwy. Podobnie jak `session_id` musi być wywołana przed funkcją `session_start`.


Slide 13

PHP: Hypertext Preprocessor

Usuwanie sesji

```
if (isset($_COOKIE[session_name()])) {
    setcookie(session_name(), "", time()-42000, '/');
}
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Aby usunąć sesję całkowicie, należy usunąć ciasteczko z jej nazwą, które jest przechowywane po stronie użytkownika. Pierw należy sprawdzić, czy takowe istnieje, a następnie należy je usunąć, metodą podaną w poprzedniej lekcji.

Slide 14




PHP: Hypertext Preprocessor

Usuwanie danych z sesji

- `session_unset();`
- `unset($_SESSION['key']);`



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Aby usunąć wszystkie dane ustawione w sesji kiedykolwiek, należy zastosować funkcję `session_unset`. Aby usunąć konkretną zmienną należy użyć drugiej konstrukcji.



Slide 15

Ćwiczenie – tworzenie sesji

- Należy przygotować skrypt, który zapisze w sesji nasze imię oraz wyświetli 'Hello, nasze_imie' z sesji. Następnie zapisze do poprzednie zmienne nazwisko i wyświetli 'Hello, nasze_nazwisko'. Na końcu usunie całą sesję.

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Wykonać ćwiczenie, na następnej stronie znajduje się rozwiązanie ćwiczenia, nie pokazywać dopóki studenci nie skończą.

Slide 16

Rozwiązanie cz.1

```
<?php
if(session_id() == "")
    session_start();
if(!isset($_SESSION['name']))
    $_SESSION['name'] = 'Filip';
echo 'Hello, ' . $_SESSION['name'];
$_SESSION['name'] = 'Kowalski';
echo '<br />Hello, ' . $_SESSION['name'];
```

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

W pierwszej kolejności inicjalizujemy sesję. Następnie ustawiamy zmienną `$_SESSION['name']` na nasze imię oraz wyświetlamy ją. Kolejnym krokiem jest nadpisanie zmiennej naszym nazwiskiem oraz wyświetlenie jej.

Slide 17

Rozwiązanie cz.2

```
session_unset();
if (isset($_COOKIE[session_name()])) {
    setcookie(session_name(), "", time()-42000, '/');
}
?>
```

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Część końcową naszego skryptu rozpoczyna usunięcie zmiennych z sesji oraz zniszczenie ciasteczka z jej identyfikatorem.



Slide 18

B2E BUSINESS TO EDUCATION **SZCZECIŃSKI PARK** NAUKOWO-TECHNOLOGICZNY PHP: Hypertext Preprocessor

Ćwiczenie – tworzenie sesji

- Napisz skrypt, który będzie zliczał i wyświetlał wizyty na stronie. Dane musi przechowywać w tablicy sesji. Korzystając z tablicy \$_GET oraz funkcji header do naszego licznika dodajmy link, który go zeruje.

DAILY PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Wykonać ćwiczenie, na następnej stronie znajduje się rozwiązanie ćwiczenia, nie pokazywać dopóki studenci nie skończą.

Slide 19

B2E BUSINESS TO EDUCATION **SZCZECIŃSKI PARK** NAUKOWO-TECHNOLOGICZNY PHP: Hypertext Preprocessor

Rozwiązanie

```
<?php
$url = (@$_SERVER["HTTPS"] == "on") ? "https://" : "http://";
if ($_SERVER["SERVER_PORT"] != "80"){
    $url =
        $_SERVER["SERVER_NAME"].":".$_SERVER["SERVER_PORT"].$
        _SERVER["REQUEST_URI"];
} else {
    $url =
        $_SERVER["SERVER_NAME"].$ _SERVER["REQUEST_URI"];
}
```

DAILY PHP: Hypertext Preprocessor - poziom średnio zaawansowany

W pierwszej kolejności potrzebujemy adres url naszej strony, będzie nam on potrzebny do zerowania licznika.

Slide 20

B2E BUSINESS TO EDUCATION **SZCZECIŃSKI PARK** NAUKOWO-TECHNOLOGICZNY PHP: Hypertext Preprocessor

Rozwiązanie



```
$url = str_replace("?clean=1", "", $url);
if(session_id() == "")
    session_start();
if(!isset($_SESSION['count']))
    $_SESSION['count'] = 0;
else
    $_SESSION['count'] = $_SESSION['count'] + 1;
```

DAILY PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Następnie czyścimy z niego zmienne z tablicy \$_GET, w przeciwnym wypadku zerowalibyśmy licznik za każdym razem. Kolejną rzeczą jest implementacja samego mechanizmu zliczającego i przechowującego dane w sesji.




Slide 21

PHP: Hypertext Preprocessor



Rozwiązanie

```
if(isset($_GET['clear']) and $_GET['clear']){  
    $_SESSION['count'] = 0;  
    header('Location: ' . $_url);  
}  
echo 'Odwiedzasz nas ' . $_SESSION['count'] . ' raz!';  
echo '<br /><a href=?clear=1?>Zeruj</a>';  
>
```

PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Na koniec potrzebna jest nam akcja, która w związku ze zmienną \$_GET wyczyści nam licznik i odświeży stronę. Kolejną rzeczą jest wyświetlenie naszego licznika oraz linku do jego zerowania.

Slide 22

PHP: Hypertext Preprocessor

Podsumowanie

- Do czego stosować sesje internetowe?

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Sesje HTTP służą głównie do systemów logowania oraz przechowywania danych globalnych lub konfiguracyjnych. Są bezpieczniejsze od ciasteczek jeśli chodzi przechowywanie danych ze względu na ich umiejscowienie (serwer). Opierają się jednak o system ciasteczek, co może spowodować brak możliwości dostania się do odpowiedniej sesji.

Slide 23

PHP: Hypertext Preprocessor

Dziękujemy za uwagę

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

9.5.3 Ćwiczenia

Ćwiczenia zostały przedstawione na slajdach. Podczas prowadzenia zajęć należy wykonywać je razem z uczniami.

9.5.4 Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą posiadać wiedzę o mechanizmie sesji oraz będą potrafili wykorzystać go w swoich skryptach, bądź aplikacjach internetowych.

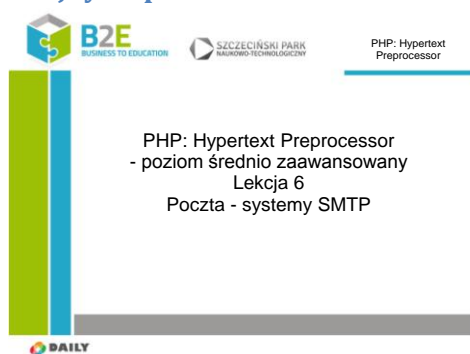
9.6 Lekcja 6 - Poczta - systemy SMTP (Systemy pocztowe w Internecie, Budowa protokołu SMTP, Komunikacja SMTP, Struktura nagłówków wiadomości, Obsługa poczty w aplikacjach PHP)

9.6.1 Cel lekcji

Celem lekcji jest zapoznanie z protokołem SMTP. Poznanie jego działania oraz nauczenie się, jak wykorzystać go w języku php, celem wysyłania poczty elektronicznej.

9.6.2 Treść - slajdy z opisem

Slide 1



PHP: Hypertext Preprocessor

PHP: Hypertext Preprocessor
- poziom średnio zaawansowany
Lekcja 6
Poczta - systemy SMTP

Słowa wstępu

Slide 2





Simple Mail Transfer Protocol – czyli SMTP

- Co to jest?
- Historia
- Zastosowanie w języku PHP


Protokół SMTP(ang. Simple Mail Transfer Protocol) – to protokół komunikacyjny, który służy jako opis standardu przesyłania poczty elektronicznej. BO raz pierwszy został opisany w roku 1982 przez Jonathana B. Postela. Często jest wymieniany razem z protokołem POP3, który służy do pobierania poczty z serwera. SMTP jest wykorzystywany w języku PHP jako warstwa do obsługi poczty elektronicznej. Możemy z niego korzystać bezpośrednio lub za pomocą np. obiektu phpmailer. Istnieje również wiele innych komponentów

wspomagających obsługę poczty elektronicznej na serwerach WWW.

Slide 3

  PHP: Hypertext Preprocessor
Systemy pocztowe w Internecie

- Gmail
- Hotmail
- Yahoo
- AOL

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Do najpopularniejszych systemów pocztowych w internecie należą te wyżej wymienione. Są dostępne z poziomu przeglądarki najczęściej po protokole HTTPS. Działają najczęściej w oparciu o protokoły IMAP(następca POP3) oraz SMTP. W niektórych istnieje możliwość dostępu bo zdefiniowanych przez nas protokołach.

Slide 4

  PHP: Hypertext Preprocessor
SMTP



- Domyślny port 25
- Wykorzystuje warstwę TCP/IP
- Adres user@domena.pl

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Protokół SMTP wykorzystuje warstwę transportową TCP, domyślnie po porcie 25. Adres pocztowy zdefiniowany jest jako część związana z domeną, czyli adres serwera na serwerze DNS, który zwraca adres IP danej domeny oraz nazwę użytkownika systemu pocztowego na danym serwerze.



Slide 5




PHP: Hypertext Preprocessor

Składnia wiadomości SMTP

```
220 domena.pl ESMTP Postfix
helo filip.domena.pl
250 domena.pl
mail from:filip@domena.pl
250 2.1.0 Ok
rcpt to:filip@domena.pl
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
This is the message body.

250 2.0.0 Ok: queued as C2BEC5B0143
quit
221 2.0.0 Bye
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Składnia wiadomości SMTP wygląda całkiem podobnie do składni żądania HTTP. Można ją podzielić na część w której znajdują się nagłówki, część z treścią oraz część kończącą wiadomość. Treść zawsze musi być zakończona znakiem kropki (.) oraz znakami powrotu karetki (CR) i znakiem przejścia do nowej linii (LF).

Slide 6



PHP: Hypertext Preprocessor

Działanie protokołu SMTP cz. 1

1. Połączenie.
2. Potwierdzenie otwarcia sesji
3. Identyfikacja hosta
4. Identyfikacja nadawcy
5. Potwierdzenie gotowości na odebranie wiadomości e-mail



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

W pierwszej kolejności nawiązywane jest połączenie TCP hosta z serwerem, do którego ma zostać wysłana wiadomość(wysyłane są trzy wiadomości, pierwsza z hosta na serwer, druga to odpowiedź serwera, trzecia prośba o otwarcie sesji). Następnie serwer SMTP wysyła wiadomość, która potwierdza otwarcie sesji. Kolejną rzeczą jest identyfikacja hosta(wiadomość z komendą EHLO). Po zidentyfikowaniu hosta, klient SMTP wysyła wiadomość MAIL służącą do identyfikacji nadawcy. W dalszej części serwer potwierdza gotowość na odebranie maila(wysyła wiadomość zawierającą komendę 250 OK).



Slide 7

PHP: Hypertext Preprocessor

Działanie protokołu SMTP cz.2

6. Wysłanie informacji o odbiorcy
7. Poinformowanie o rozpoczęciu przesyłania danych
8. Przesłanie danych
9. Zakończenie sesji - QUIT

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Klient SMTP wysyła wiadomość z informacją o odbiorcy maila (RCPT TO). Serwer akceptuje dane odpowiadając komendą: 250 OK.

Następnie klient SMTP wysyła wiadomość z komendą DATA <CRLF> informując o początku przesyłania danych maila. Serwer akceptuje komendę, odpowiadając komendą: 354. W końcu dane zostają przesłane, zakończone kropką i przejściem do nowej linii w formacie systemu windows. Klient kończy sesję wysyłając wiadomość z komendą QUIT. Serwer potwierdza koniec sesji wysyłając wiadomość z kodem: 221.

Slide 8

PHP: Hypertext Preprocessor

SMTP w PHP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Aby, PHP mogło wysyłać wiadomości via SMTP, musi mieć dostęp do serwera, który pocztowego albo do serwera, na którym jest zainstalowany i skonfigurowany protokół SMTP.



Slide 9

PHP funkcja mail

- `bool mail (string $to , string $subject , string $message [, string $additional_headers [, string $additional_parameters]])`

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Do wysyłania wiadomości e-mail bezpośrednio z serwera na którym SMTP jest skonfigurowane, można użyć funkcji mail. Funkcja zwraca wartość true, jeżeli wysyłanie wiadomości się powiodło, false w przeciwnym wypadku. Wymaga trzech argumentów: pierwszym jest adres e-mail osoby do której wysyłamy wiadomość, drugi to temat wiadomości, a trzeci jej treść. Posiada również dwa argumenty opcjonalne: pierwszy to dodatkowe nagłówki oraz drugi to dodatkowe parametry dla poleceń linii komend.

Slide 10

Ćwiczenie – wysyłania maila

- Używając skonfigurowanego serwera SMTP, wyślij maila na swoje konto. Użyj funkcji PHP ,mail'.

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Wykonać ćwiczenie

Slide 11

Rozwiązanie ćwiczenia

```
<?php
$email = 'adres@domena.pl';
$subject = 'Temat maila';
$content = 'Hello Filip';
try{
    mail($email, $subject, $content)
    echo 'Wysłano wiadomość email!';
} catch(Exception $e) {
    echo $e->message;
}
```

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

W pierwszej kolejności definiujemy adres pod który chcemy wysłać maila, treść maila oraz jego temat. Następnie próbujemy wysłać maila, jeżeli nam się uda otrzymujemy komunikat o wysłaniu, w przeciwnym wypadku treść błędu, który mówi nam o tym dlaczego się nie powiodło.



Slide 12

B2E **SZCZECIŃSKI PARK** **PHP: Hypertext Preprocessor**

Przykład obiektu klienta SMTP cz. 1

```
<?php
$CFG = new stdClass();
$CFG->SMTP->SmtServer = 'localhost';
$CFG->SMTP->SmtPort = '25';
$CFG->SMTP->SmtUser = 'me@localhost.com';
$CFG->SMTP->SmtPass = 'aaaaa';
?>
```

DAILY PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Przykład zobrazuje nam całą komunikację pomiędzy serwerem SMTP, a klientem i wyświetli w postaci tablicy. Został skompresowany na potrzeby zobrazowania niskopoziomowego mechanizmu komunikacji z serwerem oraz uszczuplony o mechanizmy zabezpieczające aplikacje. Nie należy korzystać z niego jako z gotowej aplikacji. W pierwszej kolejności tworzymy plik konfiguracyjny dla naszego obiektu umieszczamy go w osobnym pliku i wypełniamy odpowiednimi danymi., które będą nam potrzebne do wysłania maila.

Slide 13

B2E **SZCZECIŃSKI PARK** **PHP: Hypertext Preprocessor**

Przykład obiektu do klienta SMTP cz. 2

```
<?php
class MailForm{
    public static function getForm(){
        $form = '<form method="post" action=""><table>
        <tr><td>To: </td><td><input type="text" name="to" /></td></tr>
        <tr><td>From: </td><td><input type="text" name="from" /></td></tr>
        <tr><td>Subject: </td><td><input type="text" name="sub" /></td></tr>
        <tr><td colspan="2"><td><td><input type="text" name="message"></td></tr>
        <tr><td colspan="2"><td><td><input type="submit" value=" Send " /></td></tr>
        </table></form>';
        return $form;
    }
}
```

DAILY PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Potrzebujemy obiekt formularza, który będzie przetwarzany na wiadomości e-mail. Powyższy obiekt posiada jedną metodę, która zwraca nam formularz HTML w postaci ciągu znaków PHP. Umieszczamy go w osobnym pliku.

Slide 14

B2E **SZCZECIŃSKI PARK** **PHP: Hypertext Preprocessor**

Przykład obiektu klienta SMTP cz. 3

```
<?php
class SMTPClient
{
```

DAILY PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Część pierwsza klasy



Slide 15

PHP: Hypertext Preprocessor

```

function SMTPClient ($SmtServer, $SmtPort, $SmtUser, $SmtPass,
    $from, $to, $subject, $body)
{
    $this->SmtServer = $SmtServer;
    $this->SmtUser = base64_encode ($SmtUser);
    $this->SmtPass = base64_encode ($SmtPass);
    $this->from = $from;
    $this->to = $to;
    $this->subject = $subject;
    $this->body = $body;
    if ($SmtPort == ""){
        $this->PortSMTP = 25;
    } else {
        $this->PortSMTP = $SmtPort;
    }
}
    
```

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Część druga klasy, procedura, która ustawia odpowiedni parametry i dane, potrzebne do wysłania naszego maila.

Slide 16

PHP: Hypertext Preprocessor

```

function SendMail ()
{
    $stalk = array();
    $SMTP_HOST = $_SERVER['HTTP_HOST'];
    if ($SMTPPIN = fsockopen ($this->SmtServer, $this->PortSMTP))
    {
        fwrite($SMTPPIN, "EHLO " . $SMTP_HOST . "\n");
        $stalk["hello"] = fgets ($SMTPPIN, 1024);
        fwrite($SMTPPIN, "auth login\n");
        $stalk["res"] = fgets($SMTPPIN, 1024);
        fwrite($SMTPPIN, $this->SmtUser . "\n");
        $stalk["user"] = fgets($SMTPPIN, 1024);
        fwrite($SMTPPIN, $this->SmtPass . "\n");
        $stalk["pass"] = fgets($SMTPPIN, 256);
        fwrite ($SMTPPIN, "MAIL FROM: <". $this->from . ">\n");
        $stalk["From"] = fgets ($SMTPPIN, 1024);
    }
}
    
```

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Część trzecia klasy. Próbuje otworzyć połączenie z serwerem SMTP za pomocą funkcji fsockopen, a jeżeli nam się powiedzie zaczynamy się z nim komunikować. Funkcja fwrite wysyła komendy z parametrami, a funkcja fgets pobiera odpowiedź.

Slide 17

PHP: Hypertext Preprocessor

```

fwrite ($SMTPPIN, "RCPT TO: <". $this->to . ">\n");
$stalk["To"] = fgets ($SMTPPIN, 1024);
fwrite($SMTPPIN, "DATA\n");
$stalk["data"] = fgets ($SMTPPIN, 1024);
fwrite($SMTPPIN, "To: <". $this->to . ">\n\nFrom: <". $this->from . ">\n\nSubject: ". $this->subject . "\n\n\n". $this->body . "\n\n\n");
$stalk["send"] = fgets($SMTPPIN, 256);
fwrite ($SMTPPIN, "QUIT\n");
fclose($SMTPPIN);
}
return $stalk;
}
    
```

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Ostatnia część klasy. Na końcu pobieramy tablicę odpowiedzi od serwera.



Slide 18

Przykład obiektu klienta SMTP cz. 4

```
<?php
require_once('config.php');
require_once('App.php');
$app = new App($CFG);
$app->init();
?>
```

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Implementujemy główne wywołanie naszego klienta SMTP. W przykładzie można ten kod umieścić w pliku index.php w katalogu naszej aplikacji. Importujemy plik konfiguracyjny oraz klasę naszej aplikacji, która zaraz zaimplementujemy. Tworzymy jej egzemplarz i wywołujemy metodę inicjalizującą.

Slide 19

Przykład obiektu klienta SMTP cz. 5

```
<?php
class App{
    private $config;
    private $form;
    private $data;
    public function __construct($config){
        $this->config = $config;
        $this->data = new stdClass();
    }
}
```

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Pierwsza część klasy. Rozpoczynamy implementację obiektu App. W przykładzie powyżej znajdują się zmienne wewnętrzne jak dane konfiguracyjne, formularz i dane do wysłania. Plik konfiguracyjny przesyłamy za pomocą konstruktora obiektu, który wywoływany jest w momencie jego tworzenia. Funkcja aplikacji mogła by być obiektem typu singleton, lecz ze względu na brak potrzeby, mogło by to tylko zaciemnić obraz przykładu.

Slide 20

public function init(){



```
    $this->loadClasses(array("SMTPClient", "MailForm"));
    $this->form = MailForm::getForm();
    if($_SERVER["REQUEST_METHOD"] == "POST"){
        $this->getPostData();
        $this->sendMail();
    } else {
        $this->display();
    }
}
```

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Część druga implementacji klasy App. Funkcja inicjalizująca aplikację. W pierwszej kolejności, ładuje odpowiednie klasy, przekazane w argumencie jako tablica. Następnie pobiera formularz potrzebny do wysłania maila. Metoda sprawdza czy zostało coś wysłane metodą POST jeżeli tak to pobiera te dane, przetwarza je i wysyła maila. W przeciwnym wypadku wyświetla formularz.




Slide 21



PHP: Hypertext Preprocessor

```
private function loadClasses($arr){
    foreach($arr as $class)
        require_once($class.'.php');
}

private function display(){
    echo $this->form;
}
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Część trzecia implementacji klasy App. Pierwsza metoda ładuje klasy podane w tablicy jako argumenty. Druga metoda wyświetla formularz, wcześniej wczytany do obiektu.

Slide 22



PHP: Hypertext Preprocessor



```
private function sendMail(){
    $SMTPMail = new SMTPClient ($this->config->SMTP-
    >SmtServer, $this->config->SMTP->SmtPort, $this-
    >config->SMTP->SmtUser, $this->config->SMTP-
    >SmtPass, $this->data->from, $this->data->to, $this-
    >data->subject, $this->data->body);
    $SMTPChat = $SMTPMail->SendMail();
    print_r($SMTPChat);
}
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Część czwarta implementacji klasy App. Tworzymy egzemplarz wcześniej utworzonej klasy i wywołujemy jej metodę sendmail. Następnie wypisujemy tablicę z odpowiedziami od serwera SMTP.

Slide 23



PHP: Hypertext Preprocessor

```
private function getPostData(){
    if($_SERVER["REQUEST_METHOD"] == "POST")
    {
        $this->data->to = $_POST['to'];
        $this->data->from = $_POST['from'];
        $this->data->subject = $_POST['sub'];
        $this->data->body = $_POST['message'];
    }
}
```



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Ostatnia część trzecia implementacji klasy App. Funkcja przetwarzająca dane z formularza. Plik zapisujemy jako np. App.php. Nareszcie możemy zobaczyć efekt działania naszej aplikacji, pod warunkiem, że serwer SMTP jest poprawnie skonfigurowany oraz wprowadziliśmy poprawne dane w pliku konfiguracyjnym i formularzu.



Slide 24

Przykład użycia PHPMailera cz. 1

```
<?php
require_once('../class.phpmailer.php');

$mail = new PHPMailer();
$body = file_get_contents('szablon_wiadomosci.html');
```

PHP: Hypertext Preprocessor

DAILY

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

W pierwszej kolejności dołączamy klasę phpmailera do naszego skryptu, używamy `require_once` ze względu na to, że potrzebujemy dokładnie jeden egzemplarz. Następnie tworzymy egzemplarz klasy i pobieramy szablon html, bądź txt naszego maila w zależności od potrzeb.

Slide 25

Przykład użycia PHPMailera cz. 2

```
$mail->isSMTP();
$mail->Host = "mail.nazwa_domeny.pl";
$mail->SMTPAuth = true;
$mail->Port = 25;
$mail->Username = "mail.nazwa_domeny.pl";
$mail->Password = "haslo";
```

PHP: Hypertext Preprocessor

DAILY

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Ustalamy protokół na SMTP, włączamy autoryzację oraz konfigurujemy podstawowe parametry dla konta SMTP., takie jak host, port, nazwę użytkownika i hasło.

Slide 26

Przykład użycia PHPMailera cz. 3

```
$mail->From("filip@nazwa_domeny.pl");
$mail->AddReplyTo("filip@nazwa_domeny.pl", 'Filip');
$mail->Subject = "Temat";
$mail->MsgHTML($body);
```

PHP: Hypertext Preprocessor

DAILY

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

W tym miejscu ustalamy adres nadawcy, adres pod który należy wysłać odpowiedź, temat wiadomości oraz treść wiadomości, w tym przypadku wiadomość w formacie HTML.



Slide 27

Przykład użycia PHPMailera cz. 4

```
$address = "adres_odbiocy";  
// dodawanie adresów i załączników  
$mail->AddAddress($address, "Filip M");  
$mail->AddAttachment("ściezka_do_zalacznika");
```

PHP: Hypertext Preprocessor

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Możemy dodawać wiele adresów, bądź załączników do wiadomości e-mail. Należy jednak pamiętać, że wielkość załącznika jest konfigurowalna po stronie serwera, co może spowodować niemożliwość przyjęcia go przez serwer SMTP.

Slide 28

Przykład użycia PHPMailera cz. 5

```
if(!$mail->Send()) {  
    echo "Błąd phpmailera: " . $mail->ErrorInfo;  
} else {  
    echo "Wysłano!";  
}  
?>
```

PHP: Hypertext Preprocessor

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Po odpowiednim skonfigurowaniu naszego obiektu próbujemy wysłać wiadomość, w przypadku niepowodzenia otrzymuje błąd zdefiniowany w Klasie PHPMailer, w przeciwnym razie otrzymujemy komunikat o wysłaniu wiadomości.

Slide 29

Dziękujemy za uwagę

PHP: Hypertext Preprocessor

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

9.6.3 Ćwiczenia

Przykłady i ćwiczenia zawarte są na slajdach tej lekcji.

9.6.4 Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie nauczą się jak działa protokół SMTP oraz będą potrafili wykorzystać go w swoich skryptach. Nauczą się również jak zaimplementować obiektowo małą aplikację, w tym przypadku prostego klienta do wysyłania poczty elektronicznej.

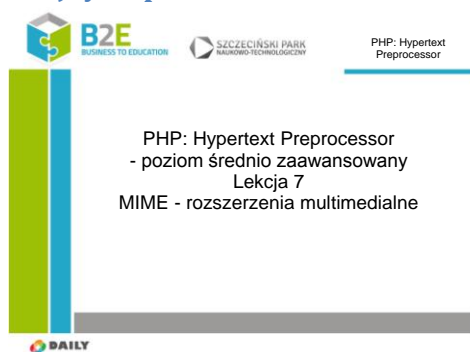
9.7 Lekcja 7 - MIME - rozszerzenia multimedialne (Struktura nagłówków MIME, Kodowanie Base64 i Quoted-Printable, Wiadomości wieloczęściowe, Obsługa poczty multimedialnej w PHP)

9.7.1 Cel lekcji

Celem lekcji jest przedstawienie protokołu MIME oraz typu multimedialnego typu internetowe, typu MIME. Poznają kodowania Base64 i Quoted-Printable. Nauczą się wysyłać pocztę multimedialną oraz poznają strukturę wiadomości wieloczęściowych.

9.7.2 Treść - slajdy z opisem

Slide 1



Slide 1 content: The slide features a green and blue vertical bar on the left with the B2E logo. The main text area contains the following information: 'PHP: Hypertext Preprocessor' (with a sub-note '- poziom średnio zaawansowany'), 'Lekcja 7', and 'MIME - rozszerzenia multimedialne'. Logos for B2E, Szczeciński Park Naukowo-Technologiczny, and Daily Group are visible in the header and footer.

Słowa wstępu

Slide 2

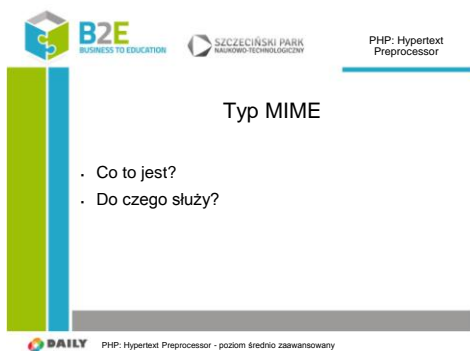


Slide 2 content: The slide features a green and blue vertical bar on the left with the B2E logo. The main text area contains the title 'MIME' and a list of bullet points: 'Co to jest?' and 'Wprowadzenie'. Logos for B2E, Szczeciński Park Naukowo-Technologiczny, and Daily Group are visible in the header and footer.

MIME (ang. Multipurpose Internet Mail Extensions) jest to standard do przesyłania poczty elektronicznej, który definiuje budowę komunikatu. Składa się z nagłówków i treści. Standardowe mail wysyła treść jako tekst zakodowany w 7-bitowym kodzie ASCII. MIME posiada kodowanie 8-bitowe, dlatego zwiększa objętość maila.



Slide 3



Typ MIME

- Co to jest?
- Do czego służy?

PHP: Hypertext Preprocessor

DAILY GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Typ MIME (nazwany od standardu MIME) – internet media type lub Content-Type (po nazwie nagłówka kilku protokołów, którego wartość jest tego typu) jest to dwuczęściowy identyfikator typu pliku w sieci, pierwsza część opisuje jego type, a druga konkretny podtyp. Np. Image/gif Image mówi, że jest to typ obrazkowy o podtypie gif, który jest zdefiniowany jako standard dla obrazów o tym rozszerzeniu. Początkowo typ ten był używany w protokole SMTP, a później zaczęto z niego korzystać również przy protokole HTTP. Służy do określania standardu przesyłanego pliku w sieci Internet.

Slide 4



Struktura nagłówków MIME

MIME-version: 1.0

PHP: Hypertext Preprocessor

DAILY GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Struktura nagłówków MIME jest identyczna ze strukturą nagłówków HTTP. Powyższy nagłówek określa wersję stosowanego standardu, w tym przypadku wersja 1.0.



Slide 5



Nagłówki MIME

Wymagane:

- MIME-version
- Content-Type

Opcjonalne:




- Content-Disposition
- Content-Transfer-Encoding
- Content-ID
- Content-Description



PHP: Hypertext Preprocessor - poziom średnio zaawansowany


W najnowszym standardzie wymagane nagłówki to wersja MIME oraz typ przesyłanej treści. Opcjonalne: Kodowanie MIME zawiera przeważnie jeden z dwóch parametrów base64 lub quoted-printable. Jest jeszcze jeden mniej popularny za to najprostszy: 8bit, nie wymagający żadnej konwersji. Nagłówek Content-ID wyznacza wiadomość właściwą, a Content-Description komentarz do wysyłanego komunikatu.

Slide 6



Składnia nagłówka Content-Type




1. Nazwa typu mediów
2. Nazwa podtypu mediów
3. Wymagane parametry (nie każdy typ tego wymaga)
4. Opcjonalne parametry (nie każdy typ tego wymaga)



PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Nagłówek Content-Type wymaga typu i podtypu mediów. Niektóre z nich wymagają parametrów, które podajemy w następnej, kolejności. Na ostatnim miejscu umieszczamy parametry opcjonalne.

Slide 7



Możliwe wartości Content-Type cz. 1

- text
- image
- audio
- video
- application
- multipart
- message



PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Część odpowiedzialna za typ, nagłówek Content-Type może przyjmować jedną z powyższych wartości. Czasami dopuszczalne, są dwie ostatnie opcje multipart oraz message. Przez użycie typu multipart, MIME pozwala, by wiadomość posiadała wiele części, z których każda może mieć określony swój własny typ MIME.



Slide 8

Możliwe wartości Content-Type cz. 2

- Podtypów MIME jest bardzo dużo, podobnie jak możliwych rozszerzeń plików, cały wykaz znajduje się w dokumentacji:
<http://www.iana.org/assignments/media-types>

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Slide 9

Możliwe wartości Content-Type cz. 3

- charset="us-ascii"

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Powyższy przykład obrazuje zastosowanie parametrów w nagłówku Content-Type, w tym przypadku jest to kodowanie znaków.

Slide 10

Kodowanie Base64

- Co to jest?
- Do czego służy?
- Jak działa?

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Kodowanie base64 to rodzaj kodowania transportowego, oznacza to, że służy do bezpiecznego przesyłania danych. Zabezpiecza dane szczególnie ze względu na uszkodzenia spowodowane niemożnością przesyłania 8 bitowych danych przez starsze maszyny, które przysyłały znaki 7-bitowe.

Kodowanie to dzieli dowolne 64 znaki na grupy po trzy bajty. Bajt posiada 8 bitów, dlatego grupa ma 24 bity. Co po podzieleniu na cztery skutkuje 6 bitowymi jednostkami. Na sześciu bitach możemy zapisać 64 różne znaki, czyli każda jednostka może przyjąć 64 wartości, stąd też nazwa kodowania. Wszystkim takim jednostką przypisywane są znaki, na podstawie

Moduł szkoleniowy SQL - Structured Query Language str. 73

Człowiek - najlepsza inwestycja

początkowo ustalonych liczb. Jeśli rozmiar wejściowego ciągu bajtów nie jest podzielny przez 3, stosowane jest dopełnienie, aż do jego podzielności przez 4.

Slide 11




 PHP: Hypertext Preprocessor
 Przykład kodowania Base64

- Hello World!




 W kodowaniu Base64:

- SGVsbG8gV29ybGQh

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Powyższy przykład obrazuje wynik kodowania ciągu znaków przy pomocy kodowania Base64.

Slide 12




 PHP: Hypertext Preprocessor
 Zastosowania kodowania Base64

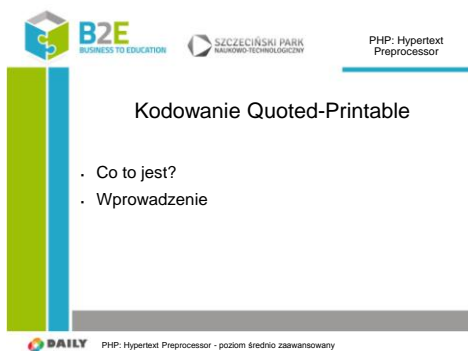
- Przesyłanie załączników binarnych w mailach
- Kodowanie haseł przy wysyłaniu protokółem SMTP

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Powyższy przykład obrazuje wynik kodowania ciągu znaków przy pomocy kodowania Base64.



Slide 13



Slide 13 content: A presentation slide titled 'Kodowanie Quoted-Printable'. It features a green vertical bar on the left and a blue vertical bar on the right. The title is centered. Below the title, there is a list of bullet points: 'Co to jest?' and 'Wprowadzenie'. At the bottom, there is a logo for 'DAILY' and the text 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany'.

Quoted-Printable – jest kolejnym kodowaniem transportowym zgodnym ze standardem MIME. Polega na tym, że pozostawia wszystkie bajty mniejsze od 128 (7-bitów), które nie są znakami sterującymi ASCII (oprócz spacji), a resztę zamienia na zakodowany szesnastkowo napis ASCII poprzedzony znakiem równości. Znak równości też jest kodowany, w celu uniknięcia wieloznaczności jako =3D. Kodowanie to jest głównie wykorzystywane w poczcie elektronicznej do kodowania przesyłanych wiadomości, które zawierają znaki narodowe z poza ASCII. Nie nadaje się do przesyłania plików binarnych ze względu na jego narzut około 225%, gdzie np. Base64 posiada tylko 33% narzutu.




Slide 14



Slide 14 content: A presentation slide titled 'Przykład kodowania Quoted-Printable'. It features a green vertical bar on the left and a blue vertical bar on the right. The title is centered. Below the title, there is a list of bullet points: 'Hello= world!' and 'Hello=3D world!'. At the bottom, there is a logo for 'DAILY' and the text 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany'.

Przykład zakodowania hello world w kodowaniu Qouted – Printable, znaki alfabetu oraz znak '!', mają kod ASCII poniżej 128, dlatego dla lepszego zobrazowania, że coś się dzieje, dodano znak równości.

Slide 15






Wiadomości wieloczęściowe

- Jak je definiować?

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Wiadomości wieloczęściowe określone jako typ multipart w nagłówku Content-Type. Dodatkowo w tym samym nagłówku muszą posiadać parametr określający separator 'boundary', którego nie wolno umieszczać w treści wiadomości. Służą one do oddzielania od siebie wiadomości.

Slide 16






Przykład wiadomości wieloczęściowej cz. 1

```
From: zuraw@domena.pl
To: czapla@domena.pl Subject: =?iso-8859-2?Q?Czaplo=zC_czy_um=F3wisz_si=EA_ze_mn=B1=3F?=
MIME Version: 1.0
Content-Type: multipart/mixed;
boundary="xxxToJestSeparator0000xxx"
This is a message with multiple parts in MIME format.
--xxxToJestSeparator0000xxx
Content-Type: multipart/alternative;
boundary="xxxToJestSeparatorZagniezdzony1111xxx"
```

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Na początku określamy adresata, odbiorcę oraz wersję MIME z której skorzystamy. Typ wiadomości ustawiony jest na multipart oraz podtyp na mixed (mieszany), który określa, że części wiadomości mogą być różnych typów. W sekcji typu umieszczamy parametr boundary, który definiuje separator w tym przypadku główny, gdyż w wiadomościach można zagnieżdżać również wiadomości.

Slide 17



Przykład wiadomości wieloczęściowej cz. 2

```
--xxxToJestSeparatorZagniezdzony1111xxx
Content-Type: text/plain; charset="iso-8859-2"
Content-Transfer-Encoding: quoted-printable
To jest tre=B6=E6 wiadomo=B6ci.
--xxxToJestSeparatorZagniezdzony1111xxx
Content-Type: text/html; charset="iso-8859-2"
Content-Transfer-Encoding: quoted-printable
```

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany



Slide 18

Przykład wiadomości wieloczęściowej cz. 3

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<HTML><HEAD>
<META http-equiv=3DContent-Type content=3D"text/html;
charset=3Diso-8859-2"></HEAD>
<BODY><FONT face=3DArial size=3D2>To jest tre=B6=E6
wiadomo=B6ci.</FONT></BODY></HTML>
--xxxToJestSeparatorZagniezdzony1111xxx--

```

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Tutaj widzimy przykład zagnieżdżonej wiadomości typu HTML.

Slide 19

Przykład wiadomości wieloczęściowej cz. 4

```

--xxxToJestSeparator0000xxx
Content-Type: image/gif; name="obrazek.gif"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="obrazek.gif"
PGh0bWw+CiAgPGhYVWQ+CiAgPC9oZWFrPgogIDxib2R5PgogICAgaP
HA+VGhpcy8pcyB0aGUg
Ym9keSBvZiB0aGUgbWVzc2FnZS48L3A+CiAgPC9ib2R5Pgog8L2h0b
Ww+CG==
--xxxToJestSeparator0000xxx--

```

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Jak widać w powyższym przykładzie po raz pierwszy użyto kodowania base64, zamiast quoted-printable. Użyto go do zakodowania obrazu typu gif.

Slide 20

Przykład wysyłania wiadomości html w PHP cz. 1

```

<?php

$to = 'stefan@domena.pl' . ' ';
$to .= 'marian@domena.pl';
$subject = 'Birthday Reminders for August';



```

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Na początek definiujemy odbiorców oraz ustalamy temat maila, ponieważ do przesłania wiadomości multimedialnej w tym przypadku w formacie html, użyjemy funkcji mail w PHP.




Slide 21

  PHP: Hypertext Preprocessor



Przykład wysyłania wiadomości html w PHP cz. 2

```
$message = '  
<html>  
<head><title>Tytuł</title></head>  
<body>  
<p>Hello world!</p>  
</body>  
</html>  
';
```

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Jako treść wiadomości definiujemy sobie kod w formacie htmla.

Slide 22

  PHP: Hypertext Preprocessor



Przykład wysyłania wiadomości html w PHP cz. 3

```
$headers = 'MIME-Version: 1.0' . "\n";  
$headers .= 'Content-type: text/html; charset=utf-8' . "\n";  
$headers .= 'From: Marek<marek@domena.pl>' . "\n";
```

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany


W nagłówkach ustalamy wersję MIME, wymagany w tym przypadku typ text/html z kodowaniem w utf8 oraz nadawcę maila.

Slide 23

  PHP: Hypertext Preprocessor

Przykład wysyłania wiadomości html w PHP cz. 4

```
if(mail($to, $subject, $message, $headers))  
    echo 'Wysłano wiadomość!'  
else  
    echo 'Wysyłanie wiadomości nie powiodło się!';  
?>
```

 PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Na końcu wysyłamy wiadomość.

Slide 24


 Slide 24 content showing logos (B2E, Szczeciński Park Naukowo-Technologiczny, DAILY) and the text "Dziękujemy za uwagę" (Thank you for attention).

9.7.3 Ćwiczenia

Jako ćwiczenia do lekcji, należy wykonać razem z uczniami przykłady znajdujące się na slajdach.

9.7.4 Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą potrafili wysyłać pocztę multimedialną np. W postaci plików html. Powinien poznać możliwe rozszerzenia plików, podstawowe nagłówki MIME oraz podstawowe kodowania transportowe. Umieć rozróżnić wiadomość wieloczęściową, od jednoczęściową.

9.8 Lekcja 8 - UML 2.1 (Diagramy czynności, Projektowanie złożonych aplikacji)

9.8.1 Cel lekcji

Celem lekcji jest zapoznanie ucznia z językiem do projektowania UML. Nauczenie go na czym polegają podstawowe diagramy oraz jak zaprojektować złożone aplikacje.

9.8.2 Treść - slajdy z opisem

Slide 1


 Slide 1 content showing logos (B2E, Szczeciński Park Naukowo-Technologiczny, DAILY) and the text "PHP Hypertext Preprocessor" and "Lekcja 8 UML 2.1".

UML jest notacją za pomocą której możliwe jest graficzne modelowanie oprogramowania. Notacja to sposób zapisu symboli, liter i znaków za pomocą której można wyrazić treść. Treścią którą będziemy omawiać jest budowa oprogramowania.



Slide 2

Czym jest UML?

- Unified Modeling Language
- Modelowanie
- Historia UML

UNIFIED MODELING LANGUAGE

PHP Hypertext Preprocessor

Lekcja 8 - UML

Rozwinięciem skrótu UML jest Unified Modeling Language czyli....
Zunifikowany Język Modelowania.

Czym jest modelowanie? Opis struktur, zachowań, interakcji, zjawisk, układu, budowy za pomocą narzędzi (notacji, diagramów..) pozwalających zobrazowanie budowy układu, sprawdzenie zachowania badanego przedmiotu. Przykładem modelu jest np. model układu słonecznego.

UML jest notacją za pomocą której możliwe jest zamodelowanie różnego rodzaju systemy. Bardzo często wykorzystywany jest do zamodelowania tworzonego oprogramowania. Pozwala na zaprojektowanie budowy oprogramowania z wykorzystaniem podejścia obiektowego, jego modułów, integracji z użytkownikiem, z systemami zewnętrznymi.

Twórcami notacji UML są G. Booch i I.Jacobson oraz J.Rumbaugh. Notacja powstała w latach 90 i jest rozwijana do dzisiaj – obecna wersja notacji to 2.2. UML jest standardem rozwijaną przez organizację OMG (Object Management Group).

Slide 3

Perspektywy implementacyjne

- przypadków użycia
- logiczna
- dynamiczna
- komponentów
- rozlokowania

PHP Hypertext Preprocessor

Lekcja 8 - UML

- przypadków użycia - definiuje funkcjonalność systemu,
- dynamiczna - prezentuje dynamikę systemu, w jaki sposób zachowuje się system,
- logiczna - dokumentuje statyczną część systemu
- komponentów - grupuje powyższe perspektywy w logicznie powiązane komponenty
- rozlokowania - specyfikuje sprzęt niezbędny do funkcjonowania określonych komponentów



Slide 4

B2E **SZCZECIŃSKI PARK** **PHP**
BUSINESS TO EDUCATION NAUKOWO-TECHNOLOGICZNY Hypertext Preprocessor

Podstawowe rodzaje diagramów UML

- Perspektywa przypadków użycia
Diagram użycia
Diagram pakietów
- Perspektywa logiczna
Diagram klas
Diagram obiektów
Diagram pakietów
- Perspektywa dynamiczna
Diagram czynności
Diagram sekwencji
Diagram pakietów
- Perspektywa rozlokowania
Diagram wdrożeniowy

DAILY Lekcja 8 - UML

Aktualnie notacja składa się z około 16 rodzajów diagramów za pomocą których możliwe jest modelowanie statycznych i dynamicznych obszarów budowy i działania systemów. Diagramy pozwalają na zaprojektowanie budowy oprogramowania, baz danych, zaprojektowanie algorytmów, określenie modułów oprogramowania, rozmieszczenie modułów oprogramowania.

Diagramy można podzielić na statyczne - przedstawiające budowę systemu oraz dynamiczne - przedstawiające np. komunikację pomiędzy elementami statycznymi. Na tej lekcji omówione zostaną podstawowe i najbardziej potrzebne rodzaje diagramów:

- Diagram użycia
- Diagram klas
- Diagram czynności
- Diagram pakietów
- Diagram wdrożeniowy

Slide 5

B2E **SZCZECIŃSKI PARK** **PHP**
BUSINESS TO EDUCATION NAUKOWO-TECHNOLOGICZNY Hypertext Preprocessor

Diagram przypadków użycia

W skład diagramu wchodzi następujące elementy:

- aktor - wykonawca akcji, funkcji
- przypadek użycia - akcja, funkcja
- granice systemu
- relacje (związki)

DAILY Lekcja 8 - UML

Diagram przypadków użycia (DPU) pozwala na zaprezentowanie funkcjonalności projektowanego systemu, jego aktorów oraz komunikację między aktorami a przypadkami użycia. Pozwala także na zaprezentowanie relacji pomiędzy aktorami i przypadkami użycia.

Diagramy nazywane w języku ang. use case są stosowane do dokumentowania wymagań wobec projektowanego systemu. Projektant ma możliwość dekompozycji wymagania na podwymagania i w procesie iteracyjnego projektowania systemu na stopniowe ich uszczegóławianie. Odpowiednio opisany zestaw przypadków użycia może stanowić dokumentację wymagań która jest podstawą do projektowania i implementacji systemu a finalnie stanowi materiał wejściowy potrzeby przy testach aplikacji.

- Przypadek użycia - konkretna

funkcjonalność, akcja którą system może wykonać w interakcji z aktorem. Przypadek użycia powinien mieć zwięzłą i jednoznaczną nazwę np. zadzwonić

- Aktor - użytkownik systemu, klient, inny system korzystający z projektowanego przez nas systemu. Aktor określany jest rzeczownikiem. Aktor reprezentuje typ użytkownika systemu, użytkownik może korzystać z wielu przypadków użycia.
- Relacja - każdy aktor na diagramie przypadków użycia powinien być związany przynajmniej z jednym przypadkiem użycia. Każdy przypadek użycia powinien być użytkowany przez co najmniej jednego aktora.
- Granice systemu - prezentują granice projektowanych systemów (na diagramie mogą wystąpić 2 lub więcej systemów których przypadki użycia należy ze sobą powiązać)
Przykłady: Przycisk dzwonić w telefonie, start w pralce, włącz silnik w samochodzie, wyloguj na facebook-u itp.

Slide 6



Na rysunku przedstawiono przykładowy diagram przypadków użycia systemu zarządzającego wypożyczalnią płyt DVD (np. osiedlowa wypożyczalnia). Projektant założył bardzo prostą funkcjonalność: Klient ma możliwość wyszukiwania filmów, wypożyczenia filmu, zwrócenia filmu

Sprzedawca ma także dostęp wyszukiwania filmów ich wypożyczenia i zwracania, ale dodatkowo może dodać nową pozycję w wypożyczalni.



Należy zauważyć że funkcjonalności są bardzo ogólne i w dalszym etapie projektowania należy je uszczegółowić - np. funkcja wypożycz film powinna sprawdzać czy dana płyta jest dostępna (nie jest wypożyczona), przydadzą się szczegóły każdej z

Moduł szkoleniowy SQL - Structured Query Language str. 82

Człowiek - najlepsza inwestycja

funkcji zwłaszcza pola służące np. do wyszukiwania, zakres danych wyświetlanych jako rezultat itp.

Slide 7


PHP
 Hypertext Preprocessor

Diagram przypadków użycia

Ćwiczenie:

Przygotuj szczegółowy diagram przypadków użycia dla biblioteki osiedlowej.

Zapewnij kompletność funkcjonalności związanej z obsługą wypożyczeń i zwrotów a także z zarządzaniem zbiorami bibliotecznymi i rezerwacjami.


 Lekcja 8 - UML

Slide 8




PHP
 Hypertext Preprocessor

Diagram klas

Elementy klasy:

- Nazwa klasy
- Atrybuty - cechy obiektów
- Operacje - czynności wykonywane na atrybutach

Klient

- imię, nazwisko
 - adres, telefon
 - e-mail, adres internetowy
 + czytać, wypożyczać, zwracać
 + czytać, wypożyczać, zwracać, rezerwować, rezerwować

Biblioteka

- nazwa, adres
 - telefon, e-mail
 - adres internetowy
 + wypożyczać, zwracać, rezerwować

Prezent

- tytuł, rok wydania
 - autor, wydawca
 - rok wydania, rok wydania


 Lekcja 8 - UML

Diagram klas jest obok diagramów przypadków użycia najczęściej stosowanym diagramem w notacji UML.

Diagram klas najlepiej odwzorowuje programowanie obiektowe. Klasa w oprogramowaniu jest definicją obiektów. Obiekt zawiera wycinek modelowanej przez oprogramowanie rzeczywistości, jest czarną skrzynką która posiada atrybuty (dane) oraz metody (funkcje) które działają na atrybutach. Obiekt może udostępniać określone metody na zewnątrz dla innych obiektów - są to metody publiczne (w programowaniu obiektowym zadeklarowane jako public na diagramie oznaczone przez +). Za pomocą udostępnionych metod świat zewnętrzny może korzystać z

funkcjonalności obiektu.

Widoczność atrybutów i metod oznaczana jest symbolami:

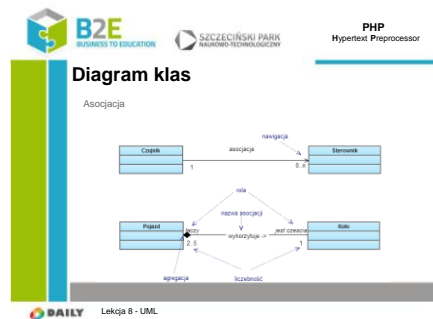
"-" - prywatne, niewidoczne na zewnątrz obiektu, dostępne tylko dla kodu obiektu

"+" - publiczne, widoczne na zewnątrz obiektu,

"#" - chronione, prywatne i dodatkowo widoczne dla klas dziedziczących

"~" - pakietowe, widoczne w pakiecie do którego należy klasa

Slide 9



Podobnie jak na diagramie przypadków użycia, diagram klas wykorzystuje związki (relacje) do obrazowania zależności między klasami. Pierwszą relacją jest asocjacja która reprezentuje dowolny związek między obiektami klasy. Asocjacja jest reprezentowana linią ciągłą i może mieć następujące cechy:

- nazwę - podkreślającą istotę związku, posiadającą strzałkę prezentującą kierunek interpretacji (np. pojazd wykorzystuje koło),
- rolę - definiującą powinność obiektu względem drugiego (np. pojazd łączy koła, koło jest częścią pojazdu),
- nawigacja - precyzująca kierunek przepływu komunikatów (np. czujnik przekazuje sygnały do sterownika),
- liczebność - definiuje zakres dopuszczalnej liczby obiektów występujących w związku (np. pojazd ma od 2 do 5 kół, koło ma 1 pojazd, czujnik jest podłączony do jednego sterownika, sterownik posiada od zera do n czujników)
- agregacja - opisuje związek całość - część. Wyróżnia się
 - agregację silną (kompozycja) (zapełniony romb) oznaczającą, że po usunięciu agregatu (obiekty grupującego) usuwane są także wszystkie segmenty (obiekty składowe),

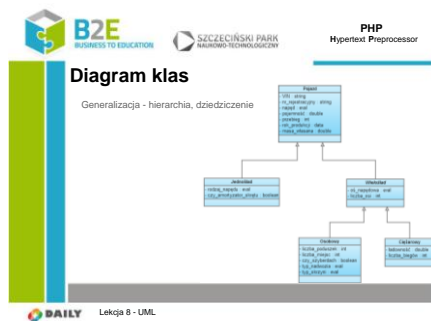
Moduł szkoleniowy SQL - Structured Query Language str. 84

Człowiek - najlepsza inwestycja

- agregację częściową (pusty romb) oznaczającą, że po usunięciu agregatu segmenty nadal mogą istnieć.

Wykorzystywanie cech jest opcjonalne i zależy od przyjętego dla danego projektu szczegółowości dokumentacji.

Slide 10



Generalizacja jest związkiem reprezentującym dziedziczenie klas. Dziedziczenie skutkuje powstaniem hierarchii klas w której najbardziej ogólna klasa jest umieszczona na samym szczycie hierarchii (korzeń). Na samym dole występują klasy nazywane liśćmi. W hierarchii występują też klasy abstrakcyjne (nazwa pisana kursywą). Klasy abstrakcyjne nie mają swoich instancji obiektów, służą natomiast do zdefiniowania wspólnych atrybutów i metod które będą dziedziczone przez klasy potomne (np. Wielośląd).

Slide 11





Slide 12

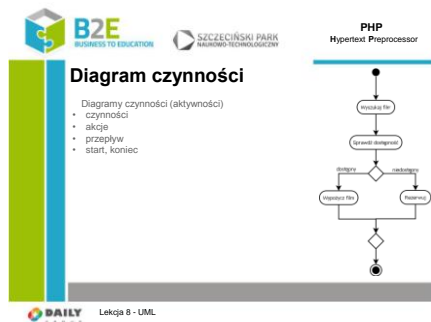
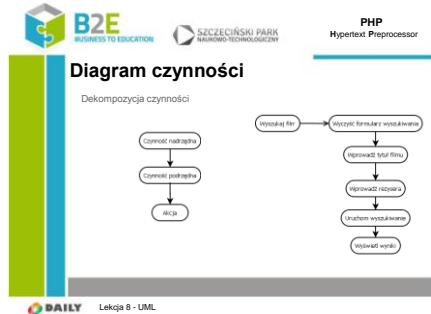


Diagram czynności służą do zobrazowania algorytmów, procesów biznesowych w projektowanym oprogramowaniu lub scenariuszy przypadków użycia. Składają się z następujących elementów:

- czynności - reprezentująca czynność złożoną, możliwą do dalszej dekompozycji na podczynności i akcje i uszczegółowienia, (np. wyszukaj film),
- akcji - elementarna jednostka zachowania, nie podlega dalszej dekompozycji, reprezentuje algorytm, metodę,
- przepływów sterowania - strzałki łączące czynności, akcje i punkty startu i zakończenia,
- punkt startu - czarne kółko,
- punkt końca - okrąg z czarną kropką w środku

Slide 13



Czynności mogą być dekomponowane. Czynność nadrzędna po uszczegółowieniu może być procesem czynności podrzędnych albo sekwencją akcji. Przykładowo, czynność "Wyszukaj film" może składać się z sekwencji czynności.



Slide 14

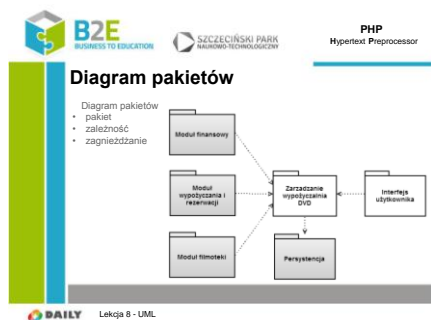


Diagram składa się z następujących elementów:

- pakiet - element grupujący inne diagramy, (ramka otaczająca elementy innych diagramów), praktyka zaleca objęcie pakietem diagramów
- zależność - łączy pakiety w logiczną strukturę, linia przerywana której kierunek wykorzystania pakietu,
- zagnieżdżanie - grupowanie innych pakietów,

Slide 15

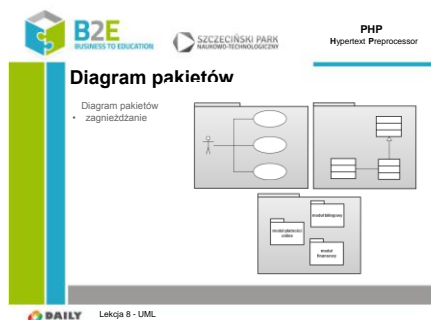
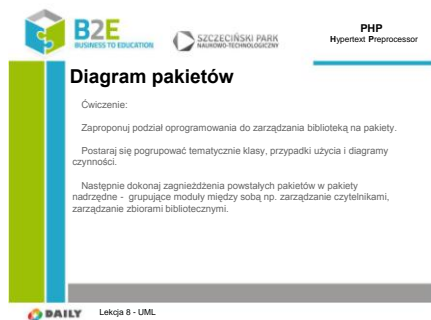


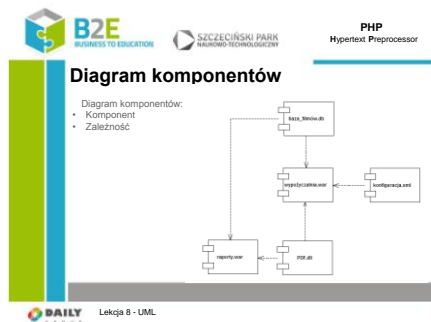
Diagram pakietów pozwala na porządkowanie i grupowanie dokumentacji. Wszystkie diagramy mogą być organizowane w pakiety (np. pakiet grupujący przypadki użycia lub diagramy klas). Pakiety mogą także grupować inne pakiety (np. pakiet dolny).

Slide 16





Slide 17

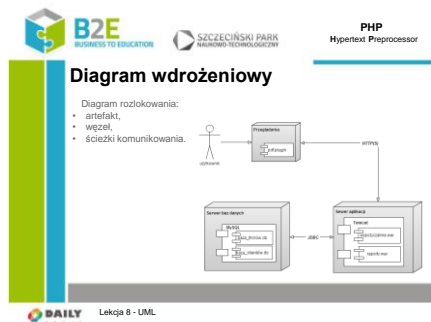


Komponentem jest izolowany, wymienny moduł oprogramowania stanowiący część systemu. Komponentami może być każdy element składowy systemu przykładowo:

- bazy danych (np. plik bazy danych, system zarządzania bazami danych),
- pliki bibliotek programowych (np. moduł do tworzenia plików PDF),
- moduły oprogramowania (np. wypożyczalnia, raporty),
- pliki (np. konfiguracyjne, graficzne).

Komponent jest reprezentowany przez symbol prostokąta z dwoma bolcami (symbolizuje interfejsu). W nowszej notacji UML dodano dodatkowe ikony komponentów pozwalających na definiowanie i nazywanie interfejsów. Zależności komponentów są reprezentowane przez linię przerywaną z grotem. Kierunek wskazuje na komponent wykorzystujący.

Slide 18



W dokumentacji często występuje potrzeba określenia fizycznego rozmieszczenia artefaktów (sztucznie wytworzonych elementów, np. komponentów) oprogramowania. Artefaktem mogą być bazy danych (np. baza_filmow.db), klasy, diagramy, biblioteki, komponenty (np. wypożyczalnia.war)

Węzłem jest fizyczny lub logiczny zasób przetwarzający na którym zainstalowano i skonfigurowano do działania artefakty (np. serwer baz danych, przeglądarka).

Ścieżki komunikowania reprezentują przepływ i kierunek przepływu informacji. Nazwa ścieżki może sugerować protokół komunikacyjny który projektuje się wykorzystać.

Slide 19



Diagram wdrożeniowy




Ćwiczenie:

Zaproponuj diagram wdrożeniowy dla oprogramowania zarządzającego biblioteką.

Postaraj się rozdzielić funkcje serwera aplikacji od serwera baz danych, umieść w węzłach odpowiednie komponenty.


 Lekcja 8 - UML

Slide 20



Pytania sprawdzające

1. Co to jest UML? Co oznacza skrót? Do czego służy?
2. Wymień min 5 diagramów UML. Jakiej masz perspektywy projektowania oprogramowania?
3. Jakim diagramem zobrazujesz wymagania na system informatyczny? Narysuj przykład diagramu i omów go.
4. Jakim diagramem zobrazujesz model obiektowy? Narysuj przykład diagramu prezentującego dziedziczenie i omów jego elementy.
5. Jakim diagramem zaprezentujesz algorytm lub proces który należy zaimplementować w systemie? Narysuj przykład diagramu i omów go.
6. Jakim diagramem zorganizujesz i pogrupujesz inne diagramy?
7. Jaki diagram służy do przedstawienia fizycznego rozmieszczenia elementów oprogramowania?
8. Do czego służy diagram komponentów?

 Lekcja 8 - UML

9.8.3 Ćwiczenia

Ćwiczenia dla tej jednostki lekcyjnej zostały umieszczone na ostatnim slajdzie.

9.8.4 Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą potrafili samodzielnie zaprojektować aplikację w języku UML oraz stworzyć podstawowe diagramy aplikacji.

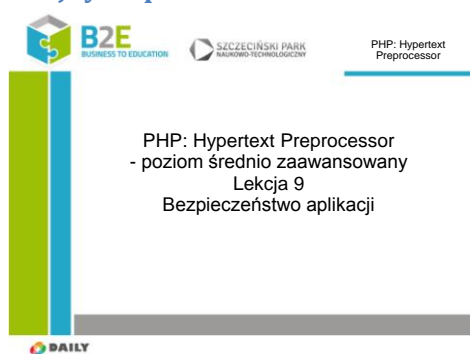
9.9 Lekcja 9 - Bezpieczeństwo aplikacji (Rodzaje ataków na portale internetowe, Testowanie poszczególnych zagrożeń, Zabezpieczanie aplikacji)

9.9.1 Cel lekcji

Celem lekcji jest poznanie rodzajów ataków na portale internetowe, opisanie możliwości testowania zagrożeń oraz podanie podstawowych technik do zabezpieczania aplikacji.

9.9.2 Treść - slajdy z opisem

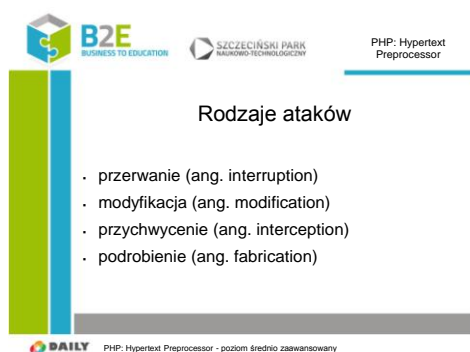
Slide 1



Slide 1 content: The slide features a green and blue vertical bar on the left. At the top, there are logos for B2E, Szczeciński Park Naukowo-Technologiczny, and PHP: Hypertext Preprocessor. The main text in the center reads: "PHP: Hypertext Preprocessor - poziom średnio zaawansowany", "Lekcja 9", and "Bezpieczeństwo aplikacji". A small DAILY logo is at the bottom left.

Słowa wstępu

Slide 2



Slide 2 content: The slide features a green and blue vertical bar on the left. At the top, there are logos for B2E, Szczeciński Park Naukowo-Technologiczny, and PHP: Hypertext Preprocessor. The main text in the center reads: "Rodzaje ataków". Below this, there is a bulleted list: "przerwanie (ang. interruption)", "modyfikacja (ang. modification)", "przechwycenie (ang. interception)", and "podrobienie (ang. fabrication)". A small DAILY logo is at the bottom left.

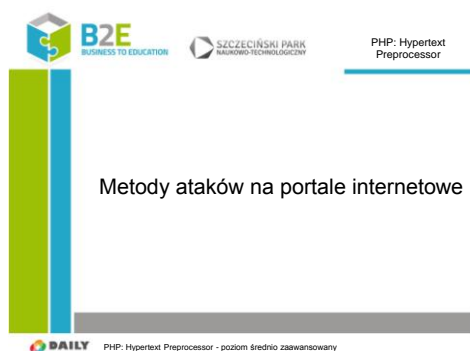
Największą zaletą aplikacji internetowych jest ich powszechność i ogólnodostępność. Zaleta ta jednak może zamienić się w wadę, ponieważ, czym więcej osób ma do czegoś dostęp, tym większe ryzyko związane z atakami. Nowe technologie sieciowe, bądź te, które są jeszcze w fazie rozwoju (do których należy PHP), narażone są na luki we własnej implementacji, bądź ich nie właściwe stosowanie. Jest wiele ataków na systemy komputerowe, jednak my zajmiemy się tylko częścią związaną z bezpośrednim dostępem do aplikacji, czyli atakami na portale internetowe. Podstawowymi typami ataków na portale są:

- Przerwanie – polega na przerwaniu

świadczenia usług przez serwer danej aplikacji np. DoS lub DDoS

- Modyfikacja – wprowadzenie zmian w aplikacji przez osoby nieupoważnione do wykonania takowych zmian
- Przechwycenie – uzyskanie kontroli nad aplikacją, bądź jej częścią lub dostęp do obszarów/zasobów, do których nie powinny mieć dostępu osoby nieuprawnione
- Podrobienie – wprowadzenie fałszywych danych do aplikacji

Slide 3



Protokół HTTP (ang. Hypertext Transfer Protocol) – to najpopularniejszy protokół wykorzystywany w sieciach komputerowych. Mówiąc prostym językiem jest to ścisły schemat postępowania(zbiór zasad) dla urządzeń komunikacyjnych w jaki sposób przysyłać dokumenty hipertekstowe. Rozpowszechniony dzięki sieci WWW(ang. World Wide Web), za jego pomocą odbywa się przysyłanie danych oraz komunikacja w większości aplikacji internetowych.



Slide 4

Atak brute-force

- Co to jest?
- Wykorzystanie bazy danych
- Metody zapobiegawcze

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Brute – force to najprostszy rodzaj ataku na systemy komputerowe, w tym portale internetowe. Polega znalezieniu formularza dostępu do aplikacji, konta użytkownika, bądź bazy danych, często też znalezieniu loginu i wpisywaniu wszystkich możliwych kombinacji haseł. W wersji ulepszonej z racji tego, że dla lepszego zapamiętania hasła ludzie stosują naturalne ciągi znaków jak wyrazy, korzysta się z zewnętrznej bazy danych haseł lub słowników np. imion.

Jest wiele metod zapobiegawczych tego typu atakom. Do najważniejszych zaliczamy w przypadku formularza logowania, który jest ogólnodostępny: losowe generatory haseł, określona z góry złożoność hasła, podpisywanie w aplikacji użytkownika inną nazwą, niż jego login (np. imieniem i nazwiskiem), blokowanie formularza dostępu po określonej ilości prób wpisania błędnego hasła (np. po trzech), systemy wymuszające zmianę hasła, co jakiś czas. W przypadku paneli administracyjnych oraz aplikacji do obsługi baz danych typu phpmyadmin, oprócz wyżej wymienionych zabezpieczeń możemy głęboko ukryć link oraz przypisać określony adres ip dostępu do panelu, bądź zezwolić na dostęp tylko z hosta lokalnego, a nie z zewnątrz. Jest jeszcze wiele innych metod, na przykład autoryzacja innych adresów IP drogą mailową lub smsesową.



Slide 5

Atak 'Głębokie ukrycie'

- Co to jest?
- Metody zapobiegawcze

PHP: Hypertext Preprocessor

DAILY GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Głębokie ukrycie to termin dotyczący bezpieczeństwa dokumentów na portalach internetowych. Umieszczamy dokumentu pod adresem, którego nie jest w stanie wpisać człowiek oraz nie indeksujemy go w wyszukiwarkach, w czym pomocny jest htmlowy `rel="nofollow"` lub ustawienia pliku robots. Metoda ataku polega na znalezieniu niezabezpieczonej ścieżki i dotarciu do dokumentów. Prosty, a mocnym zabezpieczeniem jest użycie pliku .htaccess i zabronienie dostępu osobom z zewnątrz lub autoryzacji po http. Możemy również zabezpieczyć dostęp z poziomu systemu operacyjnego np. dla systemów unixowych, zmieniamy właściciela katalogu na serwer apache, który ma wszystkie uprawnienia i zabieramy wszelkie uprawnienia innym użytkownikom.

Slide 6

Account lockout

- Co to jest?
- Metody zapobiegawcze

PHP: Hypertext Preprocessor

DAILY GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Atak polega na blokowaniu kont w aplikacjach internetowych, dlatego używając wcześniej wymienionego zabezpieczenia, które blokuje dostęp do konta, warto zadbać o prostą i szybką możliwość odblokowania konta na przykład drogą mailową, bądź smsową, czy w przypadku banków i innych instytucji zaufania publicznego, telefoniczną lub za pomocą wydrukowanych wcześniej kodów.



Slide 7

B2E **SZCZECIŃSKI PARK** **PHP: Hypertext**
BUSINESS TO EDUCATION NAUKOWO-TECHNOLOGICZNY Preprocessor

Web parameter tampering

- Co to jest?
- Metody zapobiegawcze

DAILY PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Metoda polega na przekazywaniu metodą GET lub POST, parametrów które różnią się od tych dostępnych w aplikacji. Użytkownik, może mieć w ten sposób dostęp do funkcji lub danych, do których nie powinien go mieć. Warto, więc zwrócić uwagę na to co przekazujemy ww. metodami oraz pomyśleć o szyfrowaniu parametrów, jak i ich walidowaniu. Warto też zwrócić uwagę na to w jaki sposób nasze strony komunikują się z poziomu JS -> PHP, jeżeli wykorzystujemy metody AJAX-owe oraz czy nasza aplikacja, nie wypisuje bezpośrednio tego co podajemy jako parametry, w przeciwnym wypadku możemy przekazać fragment kodu PHP, poprzez parametr i narobić szkód na portalu.

Slide 8

B2E **SZCZECIŃSKI PARK** **PHP: Hypertext**
BUSINESS TO EDUCATION NAUKOWO-TECHNOLOGICZNY Preprocessor

Path i information disclosure

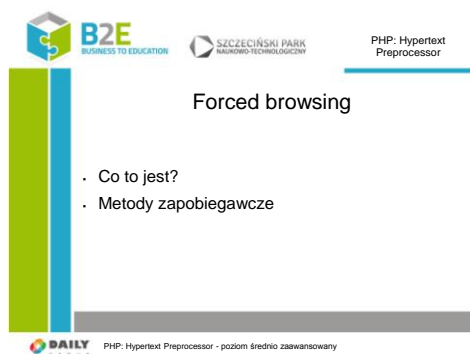
- Co to jest?
- Metody zapobiegawcze

DAILY PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Jest to rodzaj ataku, który sam w sobie jest nieszkodliwy, aczkolwiek może przynieść wiele korzyści osobie, która próbuje się włamać na naszą stronę. Dotyczy on znalezienia informacji o umiejscowieniu naszej aplikacji na serwerze, jego ustawień lub innych ważnych informacji. Informacje te można znaleźć przy błędach aplikacji oraz w obiektach konfiguracyjnych. Zapobiegać im można na wiele sposobów np. w środowisku produkcyjnym wyłączamy wyświetlanie na ekran błędów aplikacji, piszemy własną obsługę błędów, debuggujemy aplikację, tworzymy loadery do zasobów takich jak pliki js, obrazki, arkusze styli oraz innych dokumentów, które nie wyświetlają ścieżek do zasobów i informacji o

błędach w przypadku nie istnienia zasobu.

Slide 9



Slide 9 content includes logos for B2E, Szczeciński Park Naukowo-Technologiczny, and DAILY. The main text reads 'Forced browsing' followed by a bulleted list: 'Co to jest?' and 'Metody zapobiegawcze'. At the bottom, it says 'PHP: Hypertext Preprocessor - poziom średnio zaawansowany'.

Atak polega na spreparowaniu adresu url, w celu wykonania operacji, bądź uzyskania informacji, do których atakująca osoba nie powinna mieć dostępu. Problem rozwiązuje odpowiednio zaimplementowany system autoryzacji lub odpowiednia architektura aplikacji np. MVC (z którego powinniśmy, korzystać w większości aplikacji). System autoryzacji powinien za każdym razem sprawdzać czy dana osoba, nie próbuje uzyskać dostępu do odpowiedniego miejsca. Aplikacja w architekturze MVC, załatwia tę sprawę tak, że wszystko jest sterowane za pomocą odpowiedniego kontrolera, który jest dostępny tylko dla zalogowanych użytkowników.



Slide 10

B2E
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

Path traversal

- Co to jest?
- Metody zapobiegawcze

DAILY
GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Atak ten polega na zmianie nazwy pliku, w adresie do zasobu, który zwróciła aplikacja. Zapobiegać możemy jemu na wiele sposobów, poprzez np. tworzenia osobnego katalogu (ścieżki) dla zasobu, który udostępniamy lub za pomocą szyfrowania nazw pliku, co utrudni identyfikację nazw pozostałych plików.

Slide 11

B2E
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

Unicode encoding

- Co to jest?
- Metody zapobiegawcze

DAILY
GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Atak ten polega na przesłaniu za pomocą formularza zakodowanego kodu, który w przypadku wykonania, w którymś momencie aplikacji, może wywołać niepożądane efekty. Bronić się przed nim można przede wszystkim, odpowiednim przetwarzaniem formularza, nigdy nie pobieramy czystych danych z formularza i operujemy na nich. Zawsze trzeba je w odpowiedni sposób przetworzyć. Używajmy również walidacji treści pól formularza oraz określonej dziedziny znaków, która jest dostępna w odpowiednim miejscu np. Przy imieniu i nazwisku, nie potrzebne są cyfry oraz znaki inne, niż znaki alfabetu.



Slide 12

B2E
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

PHP injection

- Co to jest?
- Metody zapobiegawcze

DAILY
GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Metoda polega wstrzyknięcie kodu PHP do aplikacji, a następnie poprzez odpowiednią manipulację aplikacją uruchomienie złośliwego kodu. W szczególności należy uważać przy uploadowaniu plików na serwer, zawsze trzeba walidować ich rozszerzenie oraz zawartość. Przy ich otwieraniu bezpośrednio w przeglądarce należy je przeformatować i odpowiednio wyświetlić np. jako tekst, aby przypadkiem ich nie wykonać.

Slide 13

B2E
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

SQL injection

- Co to jest?
- Metody zapobiegawcze

DAILY
GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Jest to jedna z najpopularniejszych metod ataku na serwisy internetowe, poprzez nieodpowiednie wyescapowanie treści z formularzy, bądź nieodpowiednie składanie poleceń sql. Istnieje możliwość przekazania do nich fragmentów zapytań jako parametrów, co może spowodować wykonanie nieodpowiednich akcji na bazie danych, jak np. zmianę hasła administratora, modyfikacja parametrów konfiguracyjnych lub też usunięcie ważnych danych. Jak już wcześniej wspomniano, nie należy wstawiać czystych danych z formularzy do kodu naszej aplikacji. Zalecane jest również używanie odpowiednich obiektów, bibliotek lub adapterów, jako warstwę do komunikacji z baza danych np. PHP Database Object w skrócie PDO, który świetnie nadaje się do użycia jako warstwa do obsługi bazy danych.



Slide 14

B2E **SZCZECIŃSKI PARK** **PHP: Hypertext Preprocessor**
BUSINESS TO EDUCATION NAUKOWO-TECHNOLOGICZNY

XSS – Cross Site Scripting

- Co to jest?
- Metody zapobiegawcze

DAILY PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Atak polegający na umieszczeniu na stronie skryptu np. js, który będzie powodował niepożądane akcję przez użytkowników, np. wysyłanych ich dane logowania na inny serwer lub wykonywał inne złośliwe operacje. Stosuje się go głównie w stronach, które pozwalają użytkownikom na generowanie treści. Osadzenie skryptu w stronie HTML-owej i jego wykonanie, można wyeliminować z poziomu aplikacji poprzez np. stworzenie własnego języku znaczników lub edytora graficznego do tworzenia treści, wyświetlanie treści generowanej z odpowiednim enkodowaniem lub wyszukiwaniem skryptów w zawartości treści generowanej przez użytkownika i jego wycinaniem.

Slide 15

B2E **SZCZECIŃSKI PARK** **PHP: Hypertext Preprocessor**
BUSINESS TO EDUCATION NAUKOWO-TECHNOLOGICZNY

XST – Cross Site Tracing

- Co to jest?
- Metody zapobiegawcze

DAILY PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Atak ten wykorzystuję metodę HTTP Trace, poprzez obiekty ActiveX lub Flash. Gromadzi otrzymane w odpowiedzi nagłówki HTTP w tym nagłówki autoryzacji, który zawiera zakodowane przeważnie w base64 ciasteczko z loginem i hasłem, po czym wykorzystuje je do tworzenia własnych zapytań HTTP. Obecnie najlepszym rozwiązaniem jest nie wykorzystywanie tych technologii na stronach www. Wszystko co można zrobić we flashu, można zrobić za pomocą javascriptu i mechanizmów HTML5. A jeżeli musimy używać flasha to należy zabezpieczyć proces ładowania oraz ładować go asynchronicznie.



Slide 16

XSRF – Cross Site Request Forgery

- Co to jest?
- Metody zapobiegawcze

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Metoda polega na wysyłaniu żądań do serwera za pomocą osób trzecich, które są w nim autoryzowane. Osoba wykorzystuje uprawnienia osoby, pod którą się podszywa do wykonania akcji, wymagających zezwolenie tej osoby. Aby zapobiec tym atakom należy za każdym razem wylogować się z serwisu po zakończeniu korzystania z niego. Czas sesji serwera powinien być adekwatny do jej poufności. Żądania, które wymagają szczególnej ostrożności, jak np. przelewy, powinny wymagać dodatkowej autoryzacji np. via sms. Stosować dodatkowe ukryte dane przy autentykacji np. pseudolosową liczbę.

Slide 17

Click jacking

- Co to jest?
- Metody zapobiegawcze

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Atak ten polega na przechwyceniu zdarzenia kliknięcia użytkownika oraz wykonania innej akcji niż oczekiwana przez niego w celu uzyskania poufnych informacji lub wykonania odpowiedniej akcji, np. aktywacji mikrofonu, czy kamery internetowej poprzez technologie flash. W dzisiejszych czasach wiele stron używa tzw. Likejacking, poprzez kliknięcia w obrazki lub inne komponenty strony w tajemniczy sposób lubimy, bądź udostępniamy stronę na facebooku. Zabezpieczyć się można poprzez wtyczki w przeglądarce lub odpowiednią jej konfigurację.



Slide 18

Session hijacking

- Co to jest?
- Metody zapobiegawcze

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Metoda ta polega na przechwyceniu sesji serwera danej ofiary. Możemy do tego celu użyć kodu javascript poprzez XSS, który zwróci nam ciasteczko z identyfikatorem sesji osoby zalogowanej do serwisu. Zabezpieczyć się można, korzystając z zapamiętywania adresu IP danej sesji, połączenia poprzez SSL lub enkrypcję danych przesyłanych w odpowiedziach i żądaniach. Można również zaimplementować własny mechanizm sesji, oparty np. o bazę danych.

Slide 19

Session fixation

- Co to jest?
- Metody zapobiegawcze

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Metoda ta jest metodą pochodną, poprzedniej. Po uzyskaniu identyfikatora sesji, należy znaleźć sposób na umieszczenie w odpowiednim miejscu identyfikatora sesji, czyli obejście wcześniej wspomnianych w poprzedniej metodzie zabezpieczeń, w sposób zależny od metody zabezpieczenia.

Slide 20

Phishing

- Co to jest?
- Metody zapobiegawcze

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Jest to atak wykonywany za pomocą najczęściej telefonu. Polega na podszycie się pod zaufaną osobę, instytucję lub firmę, w celu wyłudzenia poufnych danych np. loginu i hasła użytkownika. Dzwoniąc można np. podać się za firmę hostingową, po wcześniejszym sprawdzeniu dostawcy usług internetowych i poprzez odpowiedni schemat konwersacji, uzyskać odpowiednie dane. Popularność metoda ta zawdzięcza najsłynniejszemu hakerowi w historii informatyki

Kevinowi Mitnickowi, który jak twierdzi łątał ludzi, a nie hasła. Metoda ta bazuje na socjotechnice, czyli metodzie manipulacji osobami w kontakcie bezpośrednim. Zabezpieczenie przed nią jest oczywiste, nigdy nie wolno podawać żadnych danych poufnych osobom z zewnątrz, gdyż nawet jak same firmy często zaznaczają, nikt z obsługi o takie dane prosić nie będzie.

Slide 21



Ataki DoS i DDoS

- Co to jest?
- Metody zapobiegawcze

PHP: Hypertext Preprocessor

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Ataki DoS i DDoS, czyli odpowiednio Denial of Service i Disturbed Denial of Service, polegają na zablokowaniu serwera, zasypując go ogromną ilością żądań, najczęściej wykorzystuje się do tego odpowiednie oprogramowanie, np. boty lub oprogramowanie do testowania wydajności serwera. Serwera otrzymując ogromną liczbę żądań, nie jest w stanie przetwarzać kolejnych co powoduje wstrzymanie jego pracy lub po wyczerpaniu pamięci operacyjnej jego zawieszenie. Przerwa w pracy serwera w niektórych przypadkach, może spowodować ogromne straty finansowe. Nie ma skutecznej metody zapobiegania tej metodzie.



Slide 22

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

Testowanie zagrożeń

- Jak testować bezpieczeństwo swojego portalu?

DAILY PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Jest wiele narzędzi służących do testowania aplikacji internetowych, jedne są bardziej skuteczne, drugie mniej. Należy pamiętać o tym, że testują one tylko te ataki, które są im znane. Nie są one jednak w stanie przetestować wszystkich luk w aplikacjach. Do tego celu najlepszy jest człowiek, który jest istotą kreatywną i jest w stanie znajdować luki nikomu wcześniej nie znane oraz często publikować wynik swoich badań w internecie np. na swoim blogu. Dlatego najlepszą metodą testów jest połączenie trzech czynników: programy analizujące wszystkie znane luki, wiedza ludzi, którzy szukają nowych dziur w technologiach, aplikacjach oraz architekturach, jak i własny wkład, gdyż jako programiści, znamy najlepiej swoje aplikację i rozumiemy ich potencjalne słabe punkty.

Slide 23

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

Dziękujemy za uwagę

DAILY PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Słowa na zakończenie

9.9.3 Ćwiczenia

Ćwiczenia należy wykonać z każdym uczniem z osobna.

Ćwiczenie 1 Wymień i opisz trzy wybrane przez siebie ataków na portale internetowe.

Moduł szkoleniowy SQL - Structured Query Language str. 102

Człowiek - najlepsza inwestycja

Ćwiczenie 2 Opisz metody zapobiegania atakom na portale internetowe, dla wcześniej wymienionych rodzajów ataków.

Ćwiczenie 3 Utworzyć formularz zawierający jedno pola tekstowe oraz napisać skrypt, który dokona jego walidacji, do wyboru (przydziela nauczyciel):

- Pole z imieniem (tylko znaki alfabetu)
- Numer telefonu (cyfry spacje i znak +)
- Adres (tylko znaki alfanumeryczne)

9.9.4 Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą posiadać wiedzę o popularnych atakach na portale internetowe, poznają podstawowe metody prewencji oraz będą widzieć jak testować bezpieczeństwo w swoich aplikacjach.

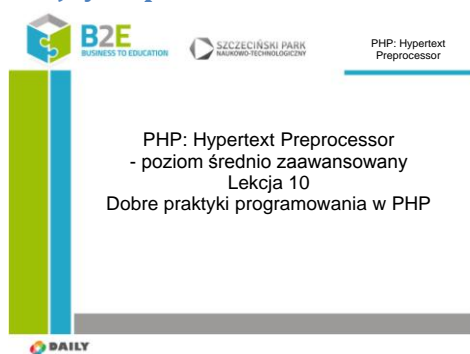
9.10 Lekcja 10 - Dobre praktyki programowania w PHP

9.10.1 Cel lekcji

Celem lekcji jest wprowadzenie ucznia w świat praktycznego podejścia do aplikacji oraz wytyczenie im dalszej drogi na rozwój w przyszłości. Pokazanie im dobrych praktyk oraz wielu przydatnych fragmentów kodu, szkieletów oraz narzędzi, które przydatne są każdemu programiście. Zaprezentowano również wzorce projektowe i obecnie najwyższy abstrakcyjnie paradygmat programowania, zwany programowaniem zorientowanym na aspekty.

9.10.2 Treść - slajdy z opisem

Slide 1



Slide 1 content area with logos (B2E, Szczeciński Park Naukowo-Technologiczny, DAILY) and text: PHP: Hypertext Preprocessor, - poziom średnio zaawansowany, Lekcja 10, Dobre praktyki programowania w PHP.

Słowa wstępu



Slide 2

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

Dobre praktyki programowania

- Wprowadzenie

DAILY GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

PHP jest językiem, w którym łatwo rozwiązuje się większość problemów. Jego prostota niesie jednak ze sobą poważną konsekwencję, kod który zaimplementowaliśmy, a który wymaga zmian jest niezrozumiały i wymaga przepisania od nowa, aby zawrzeć odpowiednie zmiany. Często też kod, który rozwiązuje podobny problem w innym projekcie, nie nadaje się do zastosowania. Należy zatem zadać sobie pytanie, czy powinniśmy po raz kolejny rozwiązywać ten sam problem, bądź też na nowo odkrywać koło, czyli pisać rzeczy, które mam w najlepszy i sprawdzony sposób zaimplementowane przez innych programistów.

Slide 3

B2E BUSINESS TO EDUCATION

SZCZECIŃSKI PARK NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

Rozszerzenia PHP

- PEAR
- PECL

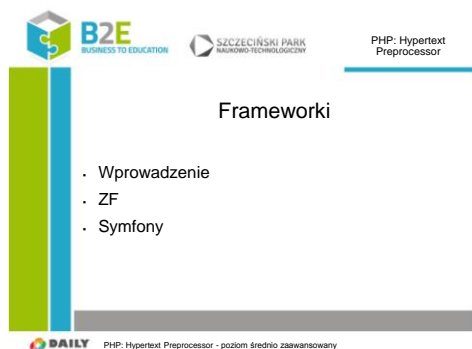
DAILY GROUP

PHP: Hypertext Preprocessor - poziom średnio zaawansowany

PEAR jest to system dystrybucji opensource'owych rozszerzeń języka PHP. Powstaje od roku 1999 i posiada wielką ilość zasobów, które ułatwiają nam pracę z językiem PHP. PECL to pierwotnie część PEAR, w przeciwieństwie do swojego rodzica, jest to repozytorium społeczności, a pakiety w nim zawarte, są pakietami nieoficjalnymi. Zanim napiszesz coś co już zostało napisane, zajrzyj do repozytorium, gdyż może to Ci pomóc zaoszczędzić mnóstwo czasu i dostarczyć źródło ze sprawdzonym rozwiązaniem.



Slide 4



Implementując aplikację internetową wykonujemy szereg operacji, które związane są z jej szkieletem oraz często powtarzalnymi rzeczami, jak autoryzacja czy systemy formularzy. Z pomocą przychodzą nam gotowe szkielety aplikacji takie jak Zend Framework czy Symfony. Jeżeli piszemy aplikację, która jest większa niż formularz kontaktowy, warto użyć gotowych rozwiązań. Szkielety te posiadają architekturę MVC oraz całą masę gotowych bibliotek np. pośredniczących w komunikacji z bazą danych, związanych z cache'owaniem, czy chociażby obsługą plików pdf. Wersje 2 powyższych frameworków, są zaimplementowane w pełni obiektowo i wykorzystują wiele wzorców poprawnego programowania, z których można czerpać świetne lekcje. Na ich stronach znajdziemy mnóstwo materiałów związanych z ich użyciem, video-tutoriale oraz wyczerpującą dokumentację np. ZF około 1300 stron formatu A4. Oprócz wyżej wymienionych frameworków, jest jeszcze wiele innych jak np. Yii, cakePHP, codeigniter czy fuel. Warto poświęcić trochę czasu na eksplorację internetu w poszukiwaniu tych zasobów, ponieważ można zaoszczędzić masę pracy i dostać kawał profesjonalnie napisanego kodu.



Slide 5

B2E
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor

Biblioteki

- Przykłady
- Composer

DAILY GROUP
PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Jeżeli nie czujesz się jeszcze na siłach, aby korzystać z wyrafinowanych narzędzi, ponieważ ich ogrom Cię przeraża lub piszesz tylko małe aplikacje dla których szkoda tracić zasoby i wydajność, możesz korzystać z frameworków jako bibliotek lub ze stricte bibliotek, których jest wiele np. TCPDF do obsługi plików pdf, czy biblioteki SMARTY do obsługi szablonów. Użytecznym narzędziem okazuje się również Composer, czyli program do zarządzania pakietami PHP.

Slide 6

B2E
BUSINESS TO EDUCATION

SZCZECIŃSKI PARK
NAUKOWO-TECHNOLOGICZNY

PHP: Hypertext Preprocessor



Programowanie obiektowe, a AOP

- Wprowadzenie
- Czy warto z niego korzystać?

DAILY GROUP
PHP: Hypertext Preprocessor - poziom średnio zaawansowany

PHP w wersji 5, zapewnia pełną obsługę programowania obiektowego, czyli najpopularniejszego obecnie programowania. Kontynuacją drogi ku wydzieleniu abstrakcji w językach programowania jest programowanie aspektowe. Implementuje się je na razie w sposób obiektowy, gdyż PHP nie zapewnia jeszcze wbudowanej obsługi aspektów. Polega ono na wydzieleniu abstrakcyjnych czynności na poziomie rozumowania człowieka np. system logów, i uniezależnienie ich od klas. Grupowanie takie zapewnia nam lepsze możliwości projektowania i bardziej zrozumiałą implementację. Zobrazujmy nasz przykład, założmy że nasz zbiór klas przy wykonywaniu jakiś czynności, zapisuje te akcje w tzw. Logach. Czy nie łatwiej by było wyodrębnić ten mechanizm z klas i przenieść go do poziomu kontenera naszej aplikacji? Zyskujemy na tym wiele czasu, nie muszą implementować go za każdym razem, choć troszkę trudniejszą. Poza tym wszelkie zmiany dokonujemy tylko raz. Co w przyszłości i przy częstych zmianach w aplikacji staje się uciążliwe.


Slide 7

PHP: Hypertext
Preprocessor



Wzorce projektowe

- Wprowadzenie
- Do czego służą?


PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Kolejną ważną rzeczą, która ułatwia życie programiście są wzorce projektowe. Są to rozwiązania problemów projektowych, które stosowane w odpowiednich przypadkach, potrafią zaoszczędzić nam wiele czasu i stworzyć łatwe do dalszego rozwijania środowisko, czy też uniknąć wielu błędów.


Slide 8

PHP: Hypertext
Preprocessor

Wzorce projektowe

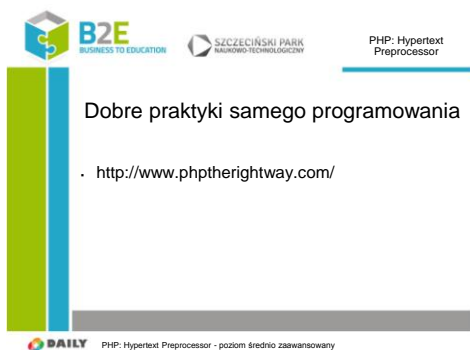
- Wprowadzenie
- Do czego służą?


PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Kolejną ważną rzeczą, która ułatwia życie programiście są wzorce projektowe. Są to rozwiązania problemów projektowych, które stosowane w odpowiednich przypadkach, potrafią zaoszczędzić nam wiele czasu i stworzyć łatwe do dalszego rozwijania środowisko, czy też uniknąć wielu błędów.



Slide 9



Slide 9 content: B2E logo, Szczeciński Park Naukowo-Technologiczny logo, PHP: Hypertext Preprocessor title, and a list item:

- <http://www.phptherightway.com/>

Jeśli chodzi o samo pisanie w języku PHP, w internecie możemy znaleźć wiele stron poświęconych dobrym praktykom programowania. Warto polecić *php the right way*, gdyż jest ona podstawą wielu publikacji na temat dobrych praktyk w języku PHP.

Slide 10

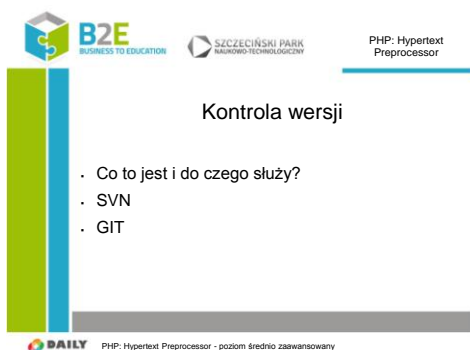


Slide 10 content: B2E logo, Szczeciński Park Naukowo-Technologiczny logo, PHP: Hypertext Preprocessor title, and a list item:

- *phpDocumentator*

Jeżeli chcemy używać naszego kodu wielokrotnie, a nie pracujemy samemu, bądź też chcemy udostępniać nasz kod dla innych programistów. Warto zadbać o dobre udokumentowanie kodu. Ułatwia ono i przyspiesza pracę nie tylko innym, ale też nam samym, kiedy po dłuższym okresie sięgamy po nasz wcześniej napisany kod. Polecanym tutaj narzędziem jest *phpDocumentator*, który tworzy dokumentację automatycznie na podstawie specjalnych komentarzy zamieszczanych w skrypcie.

Slide 11



Slide 11 content: B2E logo, Szczeciński Park Naukowo-Technologiczny logo, PHP: Hypertext Preprocessor title, and a list item:

- *SVN*

Systemy kontroli wersji lub w skrócie systemy wersjonowania, są to programy, które przechowują historię naszej pracy i ułatwiają pracę w większej grupie programistów. Do najbardziej popularnych należą *tortoise subversion* oraz *git*. Po napisaniu danego fragmentu aplikacji, możemy swobodnie wysłać ją na serwer i dołączyć do całości większej aplikacji, możemy także pobierać fragmenty napisane w przeszłości, czy przeglądać zmiany zrobione przez innych w naszym kodzie. Systemy te pozwalają także na tworzenie repozytoriów na

serwerach i udostępnianie najnowszych wersji naszych bibliotek, czy też aplikacji innym.

Slide 12



Testowanie aplikacji

- Wprowadzenie
- Testy jednostkowe
- phpUnit

PHP: Hypertext Preprocessor

DAILY PHP: Hypertext Preprocessor - poziom średnio zaawansowany

Testowanie aplikacji to jedna z ważniejszych części programowania, mamy różne rodzaje testów np. integracyjne, wydajnościowe. Jeśli chodzi jednak o najczęściej przeprowadzane testy przez samych programistów, są to testy jednostkowe. Testy jednostkowe to testy poszczególnych metod, procedur, bądź obiektów. Z pomocą przy testowaniu przychodzi nam biblioteka automatyzująca testy jednostkowe phpUnit. Warto z niej korzystać, ponieważ zaoszczędzimy sporo pracy.

Slide 13



Automatyzacja instalacji

- Wprowadzenie
- Phing

PHP: Hypertext Preprocessor

DAILY PHP: Hypertext Preprocessor - poziom średnio zaawansowany

PHP jest językiem skryptowym, stąd też nie posiada instalatorów znanych nam z systemów operacyjnych w formie pliku wykonywalnego. Są jednak narzędzia np. Phing, które pomogą nam stworzyć instalator aplikacji. Phing jest narzędziem niezależnym od platformy, wzorowanym na java-owym ANT-cie. Reguły budowania określa się w pliku xml. Po zakończonym procesie możemy zainstalować naszą aplikację, korzystając przy tym np. z rozszerzenia PEAR.

Slide 14

   PHP: Hypertext Preprocessor - poziom średnio zaawansowany


Gotowe systemy

- . CMS
- . Fora
- . Blogi
- . Sklepy
- . Platformy e-learningowe

Oprócz wcześniej wspomnianych mniejszych elementów jak biblioteki, przy tworzeniu nowej aplikacji, warto się zastanowić, czy jako podstawę do tworzenia, nie wykorzystać jakiegoś gotowego systemu, który jest udostępniany na otwartej licencji i rozwijany od wielu lat. Nasza część implementacji w tym momencie opiera się wtedy tylko, na dopisaniu odpowiednich wtyczek, komponentów, bloków lub modułów, w zależności od wybranego systemu. Wśród systemów zarządzania treścią mamy ogromny wybór, polecić tu jednak można system Joomla, czy Drupal, które należą do najbardziej rozbudowanych. Jeśli chodzi o fora internetowe napisane w języku PHP, szczególną popularnością cieszy się phpBB, z dużym wyborem darmowych szat graficznych i dodatków. Wordpress jest bezkonkurencyjnym systemem, jeśli chodzi o blogi internetowe, choć swój rozwój kieruje w stronę systemu CMS, pewnie z uwagi na prostotę i wygodę zarządzania tym systemem. Jeśli chodzi o darmowe sklepy wykonane w technologii PHP, z racji ogromnej ilości pluginów, warto polecić prestashop. Jeśli chodzi o platformy e-learningowe nie ma za dużego wyboru, zgodność ze standardami i wysoki

poziom funkcjonalności, zapewnia tylko system Moodle.

Slide 15



Slide 15 content featuring logos for B2E, Szczeciński Park Naukowo-Technologiczny, and DAILY. The text "Dziękujemy za uwagę" (Thank you for your attention) is centered. A footer line reads "PHP: Hypertext Preprocessor - poziom średnio zaawansowany".

9.10.3 Ćwiczenia

Ćwiczenie 1 Zapoznać się z dokumentacją wybranego przez siebie, nowego (nieznanego) narzędzia, frameworku lub biblioteki napisanego w języku php.

Ćwiczenie 2 Wykonać jeden lub więcej podstawowych przykładów (tutoriali) dostępnych na stronie wybranej technologii lub w jej dokumentacji.

9.10.4 Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą mieli szerszy pogląd na temat praktycznego wykorzystania języka PHP w przyszłych projektach. Będą wiedzieli, czym są wzorce projektowe, jakie narzędzia i biblioteki warto poznać oraz jakie gotowe systemy proponuje im społeczność programistów języka PHP. Lekcja ma też na celu zachęcić ich do obrania własnej ścieżki rozwoju i dalszej pracy z językiem PHP.