Универзитет у Новом Саду

ТЕХНИЧКИ ФАКУЛТЕТ „Михајло Пупин“

Зрењанин

Наставни предмет: Веб Програмирање

Школска година 2024/2025

Предметни наставник: проф. др Љубица Кази

Предметни асистент: МСц Вук Амижић

**ПРИПРЕМА ЗА РЕАЛИЗАЦИЈУ ЧАСА ВЕЖБИ**

**BROJ 8**

**ЦИЉ ВЕЖБЕ**

Циљ вежбе је да студенти овладају интеграцијом готових веб сервиса са интернета, повезивањем са истом или различитом технологијом, укључујући имплементацију веб сервиса и креирање рест сервиса у PHP-у.

**ТЕОРИЈСКИ УВОД СА ПРИМЕРИМА И ОБЈАШЊЕЊИМА**

REST веб сервиси у PHP-у представљају начин креирања интерфејса који омогућава комуникацију између различитих софтверских система преко HTTP протокола. У оквиру ових сервиса, сервер обрађује захтеве послате са клијентске стране и враћа податке у стандардизованом формату, најчешће у облику JSON-а. Оваква архитектура се ослања на јасну поделу одговорности између клијента и сервера, што омогућава лакше одржавање и проширење система.

PHP се у оквиру REST приступа користи за обраду захтева путем различитих HTTP метода као што су GET, POST, PUT и DELETE. Приликом обраде ових захтева, серверска логика се усмерава на рад са ресурсима, као што су подаци у бази, и одговара са прецизно дефинисаним кодовима стања, како би клијент могао да препозна исход обраде.

На клијентској страни, JavaScript се користи за слање асинхроних захтева према PHP серверу. То се најчешће реализује преко fetch функције или XMLHttpRequest објекта. Асинхроно слање захтева омогућава веб апликацијама да комуницирају са сервером без потребе за поновним учитавањем странице, што значајно унапређује корисничко искуство. Клијент обрађује одговор сервера и по потреби ажурира садржај странице динамички, на основу добијених података.

PHP може да комуницира са серверима који користе ASP.NET, као и обрнуто. У контексту веба, оно што омогућава ову комуникацију нису конкретно језици (PHP или C#), већ протоколи и стандарди као што су HTTP и формати за размену података као што је JSON.

Фронт-енд у PHP-у, односно PHP скрипта која се извршава на серверу, може слати HTTP захтеве ка другом серверу на коме ради ASP.NET. На пример, PHP може да користи библиотеке као што су cURL или file\_get\_contents() за слање захтева ка ASP.NET REST API-ју и пријем одговора. Одговор се затим може обрадити и приказати кориснику.

Исто важи и у обрнутом смеру – ASP.NET сервер може слати захтеве ка PHP серверу. Једини услов је да оба сервиса користе заједнички протокол комуникације, као што је HTTP, и да размењују податке у разумљивом формату, најчешће JSON или XML.

**РЕАЛИЗОВАН ПРИМЕР**

**Пример 1 – РЕСТ веб сервис**

У прилогу испод, налази се пример REST веб сервиса у оквиру PHP-a:

Изворни код за index.php налази се у прилогу испод:

|  |
| --- |
| <!DOCTYPE *html*>  <html *lang*="sr">  <head>      <meta *charset*="UTF-8">      <title>CRUD Асинхрони захтеви</title>      <link *rel*="stylesheet" *href*="index.css">  </head>  <body>      <h1>PHP + JavaScript CRUD симулација</h1>      <div *class*="dugmad">          <button *class*="crud-dugme" *data-akcija*="create">Креирај</button>          <button *class*="crud-dugme" *data-akcija*="read">Прочитај</button>          <button *class*="crud-dugme" *data-akcija*="update">Ажурирај</button>          <button *class*="crud-dugme" *data-akcija*="delete">Обриши</button>      </div>      <div *id*="rezultat"></div>      <script *src*="index.js"></script>  </body>  </html> |

Објашњење изворног кода:

Изворни код за index.js налази се у прилогу испод:

|  |
| --- |
| *document*.*addEventListener*("DOMContentLoaded", () *=>* {  *const dugmad = document.querySelectorAll(*".crud-dugme"*)*;  *const rezultat = document.getElementById(*"rezultat"*)*;  *dugmad*.*forEach*(dugme *=>* {  *dugme*.*addEventListener*("click", () *=>* {  *const akcija = dugme.dataset.akcija*;  *fetch*("servis.php", {                  method: "POST",                  headers: {                      "Content-Type": "application/json"                  },                  body: *JSON*.*stringify*({ akcija: akcija })              })              .*then*(response *=>* *response*.*json*())              .*then*(data *=>* {  *rezultat*.*textContent* *=* `Одговор: ${*data*.*odgovor*}`;              })              .*catch*(error *=>* {  *rezultat*.*textContent* *=* "Грешка при комуникацији са сервером.";  *console*.*error*("Грешка:", error);              });          });      });  }); |

Објашњење изворног кода:

Ово је JavaScript фајл који управља понашањем дугмади. Када се неко од четири дугмета кликне, шаље се асинхрони захтев ка servis.php, и резултат се приказује на страници – без поновног учитавања.

**Садржи:**

* Слушање клика на дугмад
* Употребу fetch() за слање POST захтева
* Слање JSON података са акцијом (create, read, итд.)
* Приказ одговора сервера у елементу #rezultat

Изворни код за servis.php налази се у прилогу испод:

|  |
| --- |
| <?php  header('Content-Type: application/json');  *// Читање захтева*  *$input* *=* json\_decode(file\_get\_contents('php://input'), true);  *$akcija* *=* *$input*['akcija'] *??* null;  *// Пример логике за симулацију CRUD*  *switch* (*$akcija*) {  *case* 'create':  *$odgovor* *=* "Нови податак је креиран.";  *break*;  *case* 'read':  *$odgovor* *=* "Податак је прочитан.";  *break*;  *case* 'update':  *$odgovor* *=* "Податак је ажуриран.";  *break*;  *case* 'delete':  *$odgovor* *=* "Податак је обрисан.";  *break*;  *default*:  *$odgovor* *=* "Непозната акција.";  *break*;  }  *// Враћање JSON одговора*  echo json\_encode(['odgovor' *=>* *$odgovor*]); |

Објашњење изворног кода:

Ово је PHP фајл који ради као „мини сервер“. Он прима захтеве са клијентске стране (из index.js), обрађује их и враћа JSON одговор. Не врши праву базу података, али симулира одговоре за сваку CRUD операцију.

**Садржи:**

* Читање JSON захтева са php://input
* Препознавање акције: create, read, update, delete
* Враћање JSON одговора назад клијенту

**Пример 2 - РЕСТ веб сервис**

У прилогу испод, налази се пример REST веб сервиса у оквиру PHP-a:

Изворни код за index.php налази се у прилогу испод:

|  |
| --- |
| <!DOCTYPE *html*>  <html *lang*="sr">  <head>      <meta *charset*="UTF-8">      <title>Унос података о особи</title>      <link *rel*="stylesheet" *href*="index.css">  </head>  <body>      <h1>Форма за унос података о особи</h1>      <form *id*="osoba-forma">          <input *type*="text" *id*="ime" *placeholder*="Име" *required*>          <input *type*="text" *id*="prezime" *placeholder*="Презиме" *required*>          <input *type*="number" *id*="godina" *placeholder*="Година рођења" *required*>          <button *type*="submit">Пошаљи</button>      </form>      <div *id*="rezultat"></div>      <script *src*="index.js"></script>  </body>  </html> |

Објашњење изворног кода:

Структура ХТМЛ-а се састоји од семантичких елемената и форме за особу, где се уносе подаци попут имена, презимена и година особе, и укључује дугме за слање форме.

**Напомена: форма не садржи тип акције и путању за слање података, јер се тај део одвија као РЕСТ.**

Изворни код за index.css налази се у прилогу испод:

|  |
| --- |
| body {      font-family: sans-serif;      background-color: #f9f9f9;      padding: 40*px*;      text-align: center;  }  h1 {      color: #333;      margin-bottom: 30*px*;  }  form {      display: flex;      flex-direction: column;      align-items: center;      gap: 15*px*;      margin-bottom: 20*px*;  }  input {      padding: 10*px*;      font-size: 16*px*;      width: 250*px*;  }  button {      padding: 10*px* 20*px*;      font-size: 16*px*;      background-color: #0078d7;      color: white;      border: none;      border-radius: 5*px*;      cursor: pointer;  }  button*:hover* {      background-color: #005fa3;  }  *#rezultat* {      font-size: 18*px*;      color: #222;      margin-top: 20*px*;  } |

Објашњење изворног кода:

Овај изворни код представља основно стилизовање странице. Подешена је стилизација за дугмад, инпут поља, форму и наслове.

Изворни код за index.js налази се у прилогу испод:

|  |
| --- |
| *document*.*addEventListener*("DOMContentLoaded", () *=>* {  *const forma = document.getElementById(*"osoba-forma"*)*;  *const rezultat = document.getElementById(*"rezultat"*)*;  *forma*.*addEventListener*("submit", (e) *=>* {  *e*.*preventDefault*();  *const ime = document.getElementById(*"ime"*).value*;  *const prezime = document.getElementById(*"prezime"*).value*;  *const godina = document.getElementById(*"godina"*).value*;  *fetch*("servis.php", {              method: "POST",              headers: {                  "Content-Type": "application/json"              },              body: *JSON*.*stringify*({                  ime: ime,                  prezime: prezime,                  godina: godina              })          })          .*then*(response *=>* *response*.*json*())          .*then*(data *=>* {  *rezultat*.*textContent* *=* `Сервер каже: ${*data*.*odgovor*}`;          })          .*catch*(error *=>* {  *rezultat*.*textContent* *=* "Грешка при комуникацији са сервером.";  *console*.*error*("Грешка:", error);          });      });  }); |

Објашњење изворног кода:

Када се цела страница учита, покреће се функција која чека да се нешто уради са формом. Узима се форма са странице и простор где ће касније бити приказан резултат.

Када се форма пошаље, зауставља се стандардно слање и уместо тога се узимају вредности из поља за име, презиме и годину. Те вредности се пакују у JSON формат и шаљу на сервер на адресу „servis.php“ методом која не захтева освежавање странице.

Када сервер врати одговор, он се претвара у објекат и из њега се узима део који садржи поруку. Та порука се исписује у делу странице предвиђеном за резултат. Ако дође до грешке током слања или примања, уместо одговора се исписује порука о грешци и та грешка се прикаже у конзоли.

Изворни код за servis.php налази се у прилогу испод:

|  |
| --- |
| <?php  header('Content-Type: application/json');  *$input* *=* json\_decode(file\_get\_contents('php://input'), true);  *$ime* *=* *$input*['ime'] *??* '';  *$prezime* *=* *$input*['prezime'] *??* '';  *$godina* *=* *$input*['godina'] *??* '';  *if* (*$ime* *&&* *$prezime* *&&* *$godina*) {  *$odgovor* *=* "Примљени подаци: *$ime* *$prezime*, рођен(а) *$godina*.";  } *else* {  *$odgovor* *=* "Недостају неки подаци.";  }  echo json\_encode(['odgovor' *=>* *$odgovor*]); |

Објашњење изворног кода:

На почетку се подешава да одговор који скрипта шаље буде у JSON формату. То значи да ће се клијенту вратити подаци у структурираном облику, погодном за обраду у JavaScript-у.

Затим се узимају подаци који су послати у захтеву. Они долазе као сиров текст и претварају се у низ помоћу json\_decode. У тим подацима се траже вредности за име, презиме и годину. Ако неки од тих делова недостаје, уместо њих се користи празан текст.

Проверава се да ли су сва три податка присутна. Ако јесу, формира се порука у којој пише шта је стигло, спојено у један текст. Ако нешто недостаје, шаље се порука да подаци нису комплетни.

На крају, та порука се пакује у JSON формат и шаље назад као одговор.

**Напомена: део са доделом вредности имена, презимена и година може да се реализује и преко обичне if провере, где се, уколико нешто није послати, додељује празан стринг.**

**Пример 3 – Интеграција веб сервиса независног од технологије (постојећи веб сервис)**

Изворни код за index.php налази се у прилогу испод:

|  |
| --- |
| <!DOCTYPE *html*>  <html *lang*="sr">  <head>      <meta *charset*="UTF-8">      <title>Насумичан корисник</title>      <link *rel*="stylesheet" *href*="index.css">  </head>  <body>      <h1>Преузми насумичне податке о особи</h1>      <button *id*="preuzmi-dugme">Преузми особу</button>      <div *id*="rezultat"></div>      <script *src*="index.js"></script>  </body>  </html> |

Објашњење изворног кода:

Структура примера садржи фајл за стилизовање, јаваскрипт фајл, поље за резултат и дигме на које ће бити везан јаваскрипт код.

Изворни код за index.css налази се у прилогу испод:

|  |
| --- |
| body {      font-family: sans-serif;      background-color: #f0f0f0;      text-align: center;      padding: 40*px*;  }  h1 {      color: #222;  }  button {      padding: 12*px* 24*px*;      font-size: 16*px*;      background-color: #0078d7;      color: white;      border: none;      border-radius: 6*px*;      cursor: pointer;  }  button*:hover* {      background-color: #005fa3;  }  *#rezultat* {      margin-top: 30*px*;      font-size: 18*px*;      color: #333;  }  *#rezultat* img {      width: 150*px*;      border-radius: 50*%*;      margin: 15*px* 0;  }  body {      font-family: sans-serif;      background-color: #f0f0f0;      text-align: center;      padding: 40*px*;  }  h1 {      color: #222;  }  button {      padding: 12*px* 24*px*;      font-size: 16*px*;      background-color: #0078d7;      color: white;      border: none;      border-radius: 6*px*;      cursor: pointer;  }  button*:hover* {      background-color: #005fa3;  }  *#rezultat* {      margin-top: 30*px*;      font-size: 18*px*;      color: #333;  }  *#rezultat* img {      width: 150*px*;      border-radius: 50*%*;      margin: 15*px* 0;  } |

Објашњење изворног кода:

Направљена је стилизација на основу постојећих хтмл елемената попут дугмади, главног дела странице, наслова и ID-јева.

Изворни код за index.js налази се у прилогу испод:

|  |
| --- |
| *document*.*addEventListener*("DOMContentLoaded", () *=>* {  *const dugme = document.getElementById(*"preuzmi-dugme"*)*;  *const rezultat = document.getElementById(*"rezultat"*)*;  *dugme*.*addEventListener*("click", () *=>* {  *fetch*("https://randomuser.me/api/")              .*then*(response *=>* *response*.*json*())              .*then*(data *=>* {  *const osoba = data.results[0]*;  *const ime = osoba.name.first*;  *const prezime = osoba.name.last*;  *const slika = osoba.picture.large*;  *const email = osoba.email*;  *const lokacija = `${osoba.location.city}, ${osoba.location.country}`*;  *rezultat*.*innerHTML* *=* `                      <h2>${ime} ${prezime}</h2>                      <img src="${slika}" alt="Слика особе">                      <p><strong>Email:</strong> ${email}</p>                      <p><strong>Локација:</strong> ${lokacija}</p>                  `;              })              .*catch*(error *=>* {  *rezultat*.*textContent* *=* "Грешка при преузимању података.";  *console*.*error*("Грешка:", error);              });      });  }); |

Објашњење изворног кода:

Када се страница учита, поставља се слушање на дугме. Узимају се два елемента са странице — један је дугме које покреће радњу, други је простор где ће бити приказан резултат.

Када се притисне дугме, шаље се захтев ка спољном API-ју који враћа насумичне податке о некој особи. Када стигне одговор, он се претвара у објекат и узима се први резултат из добијених података.

Из тог резултата се издвајају име, презиме, слика, имејл и локација, која се састоји од града и државе. Сви ти подаци се затим спајају у HTML и убацују у предвиђено место на страници, тако да се прикажу као структура са сликом и текстом.

Ако током преузимања дође до било какве грешке, уместо података се приказује порука о грешци, а детаљи те грешке се исписују у конзоли.

**ЗАДАТАК ЗА ИЗРАДУ ОД СТРАНЕ СТУДЕНАТА**

Направити РЕСТ веб сервис на основу датог примера реализованог решења.   
Неопходно је направити форму која садржи име, презиме, године и датум неке особе. Подаци се прослеђују на веб сервис који обрађује податке. Уколико су сви подаци унети, веб сервис обавештава клијента да су подаци попуњени. У супротном, враћа поруку за грешку.