HUMAN CAPITAL
NATIONAL COHESION STRATEGY

EUROPEAN UNION
EUROPEAN
SOCIAL FUND

Server side programming and AJAX requests

Michał Okulewicz, MSc
Warsaw University of Technology
Faculty of Mathematics and Information Science
M.Okulewicz@mini.pw.edu.pl
http://www.mini.pw.edu.pl/~okulewiczm

Server side programming tasks

- We will be using Microsoft .NET WebAPI as an example of RESTful web interface
- We might be using Microsoft .NET WCF as an example of JSON based WebService
- In all the tasks please use Visual Studio 2015

Task 1

- Create an empty ASP.NET Web Project with WEB API enabled
- Take the solution of the asynchronous spreadsheet from the previous laboratories
- Design and implement a RESTful backend interface with the following URIs and METHODS
  - GET api/table
  - GET api/table/{row}/{column}
  - POST api/table/{row}/{column} + new value in request message body
  - DELETE api/rows/{id}
  - DELETE api/columns/{id}
  - PUT api/rows
  - PUT api/columns
- You may use HTTP Session – AS A MOCK DATABASE

Task 1 – enabling the Session

```
namespace Task1
{
    public class WebApiApplication : System.Web.HttpApplication
    {
        protected void Application_PostAuthorizeRequest()
        {
            HttpContext
                .Current.SetSessionStateBehavior(SessionStateBehavior.Required);
        }
        protected void Application_Start()
        {
            GlobalConfiguration.Configure(WebApiConfig.Register);
        }

    }
}
```

Task 1 – routing and mapping hints

```
config.Routes.MapHttpRoute(
    name: "RowsApi",
    routeTemplate: "api/rows/{id}",
    defaults: new { controller = "rows", id = -1 }
);

config.Routes.MapHttpRoute(
    name: "DefaultApi",
    routeTemplate: "api/{controller}/{r}/{c}",
    defaults: new { r = RouteParameter.Optional,
                 c = RouteParameter.Optional }
);
```

```
public IHttpActionResult Get(int r, int c)
{
   var data = Task4.Table.GetTable(HttpContext.Current.Session);
   if (r > -1 && r < data.Count && c > -1 && c < data[r].Count)
       return Ok(data[r][c]);
   return NotFound();
}
```

Task 2

- Take grouped numbers JSON generator from Task3 from previous labs
- Change a WebForm to a WebAPI RESTfull interface
- Enhance the HTML webpage with a form defining the number of groups and the range of values
- All the necessary data should be sent by client, without storing anything on a server side
- Perform necessary modifications of the JS scripts and prepare .NET
- Set default values for number of groups and the range of values on the server side

Task 3 [WCF Bonus]

- Replace a WebApi Controller with an AJAX-enabled WCF Service
- Create a method returning an array of the objects containing a group name and an array of integer values
- Observe the returned object in the console and make the necessary adjustments in JavaScript and .NET WCF Service code

Task 4 [WCF Bonus]

- Change the WCF web service in such a way that it will accept the number of groups as a first argument and the range of values as the second one
- Separate data generation from setting the number of groups and range of values to different methods and store the settings in the Session variables
- Perform client side and server side validation for max > min range