REACT

11 FETCH AND JQUERY AJAX CALLS

SYLLABUS

- 01 Fundamentals And Installation
- 02 Components And Props
- 03 State
- 04 Handling Events
- 05 SetState And Lifecycle
- 06 Lists And Keys
- 07 Forms
- 08 PropTypes
- 09 Refs
- 10 React Router v4
- **11 Fetch And jQuery Ajax calls**

CROSS-ORIGIN RESOURCE SHARING (CORS)

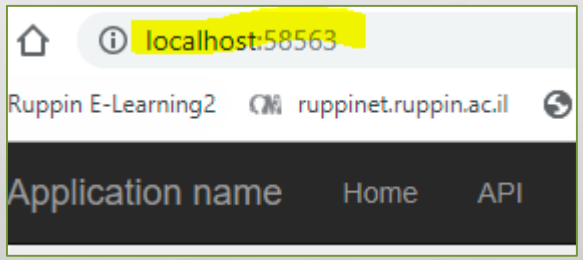
```
<script>
    function btnGetStudents() {
        alert(1);

        var url = 'http://localhost:58563/api/studentsCustom';
        fetch(url,
            {
                method: 'GET', // 'GET', 'PUT', 'DELETE', etc.
                headers: new Headers({
                    'Content-Type': 'application/json'
                }),
            }) // Call the fetch function passing the url of the
            .then((resp) => resp.json()) // Transform the data
            .then(function (data) {
                alert(data);
            })
            .catch(function (err) {
                alert(err);
            });
    }
</script>
</head>
<body>
<button onclick="btnGetStudents()">Get Students</button>
```

- כאשר נרצה לקרוא ע"י FETCH ל- WEB API מדומיין\אתר (פרויקט - SOLUTION) אחר או אחר REACT למשל . נצטרך לאפשר זאת ע"י הוספת יכולת CORS, אחרת נקבל שגיאה.
- נניח שהלקוח שלנו הוא אתר אחר עם הקוד JS הבא:

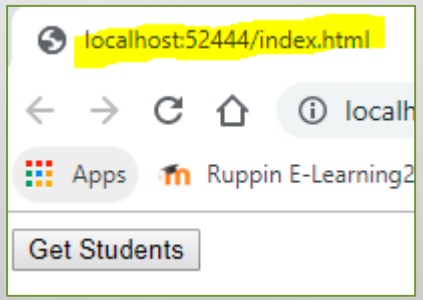
CROSS-ORIGIN RESOURCE SHARING (CORS) CONTINUE

- שימו לב שהלקוח והשרת רצים תחת דומיין (פה ה-PORT) שונה.
- נקבל את השגיאה הבאה:

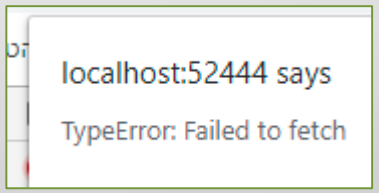
sever

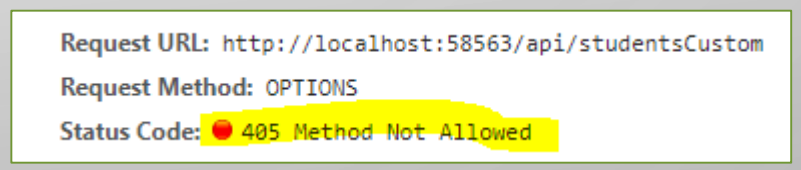


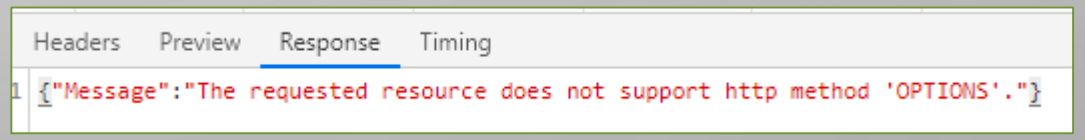
client



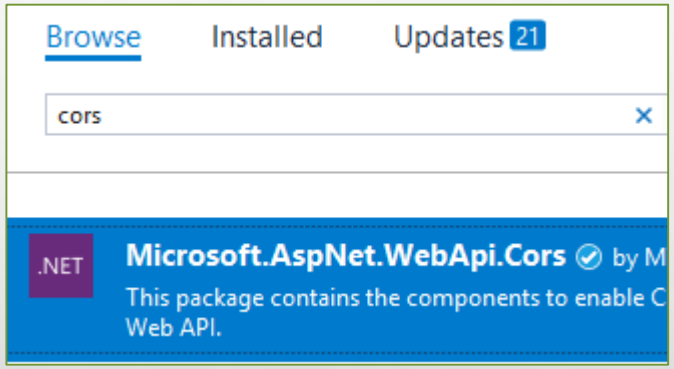
errors







CROSS-ORIGIN RESOURCE SHARING (CORS) CONTINUE



- בכדי לאפשר CORS יש צורך להתקין את הPACKAGE הבא:

- ניתן לאפשר את הCORS בכמה רמות :
- ברמת הפרויקט כולו

```
public static void Register(HttpConfiguration config)
{
    EnableCorsAttribute cors = new EnableCorsAttribute("*", "*", "*");
    config.EnableCors(cors);
```

- ברמת הקונטרולר

```
[EnableCors("*", "*", "*")]
public class StudentsCustomController : ApiController
{
```

- ברמת המתודה הבודדת.

```
[EnableCors("*", "*", "*")]
public IHttpActionResult Get(string gradeStatus = "all")
{
```

CROSS-ORIGIN RESOURCE SHARING (CORS) CONTINUE

- ניתן להגביל את המאפיינים לפתיחת ה-CORS
  - origins
  - headers
  - methods

```
EnableCorsAttribute cors =
    new EnableCorsAttribute("http://localhost:52444", "accept,Content-Type", "GET");
```

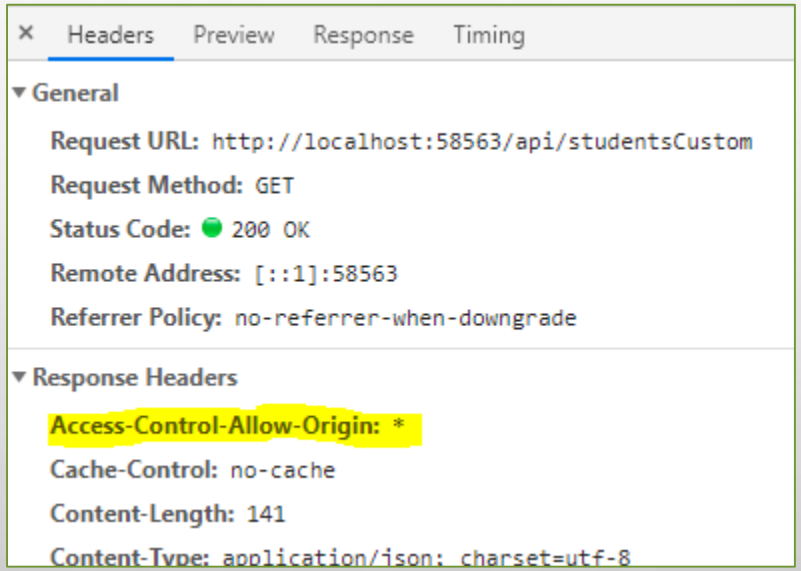
```
[EnableCors("*", "*", "*")]
public IHttpActionResult Get(string gradeSt
```

- "*" מאפשר את כל האופציות

- ניתן גם לאסור עבור רמה מסויימת ע"י [DisableCors]

```
[DisableCors]
public IHttpActionResult Get(string gradeStatus = "all")
{
```

CROSS-ORIGIN RESOURCE SHARING (CORS) CONTINUE



- ניתן לראות שה- HEADER מסוג CORS נוסף ל- RESPONSE

WEB API – WITH MODEL STUDENT CLASS

```
public class Student
{
    string id;
    double avg;
    string fullName;
    public Student() { }
    public Student(string id, double avg, string fullName)
    {
        Id = id;
        Avg = avg;
        FullName = fullName;
    }

    public string Id { get => id; set => id = value; }
    public double Avg { get => avg; set => avg = value; }
    public string FullName { get => fullName; set => fullName = value; }
    static public List<Student> students = new List<Student>
    {
        new Student("303030",92.98,"Beyonce"),
        new Student("383838",75.6,"Avi Avi"),
        new Student("404040",89.12,"Benny Benny"),
        new Student("484848",94.12,"Nir Nir")
    };
```

```
public Student GetStudent(List<Student> students,string name)
{
    foreach (var s in students)
    {
        if (s.FullName.ToLower()==name.ToLower())
        {
            return s;
        }
    }
    return null;
}
public void RemoveStudent(List<Student> students, string id)
{
    foreach (var s in students)
    {
        if (s.Id.ToLower() == id.ToLower())
        {
            students.Remove(s);
            break;
        }
    }
}
```

RESTFUL API AND FETCH GET

Needs to be change according to the actual url

```
fetch(this.apiUrl)
      .then(res => {
        console.log('res=', res);
        console.log('res.status', res.status);
        console.log('res.ok', res.ok);
        return res.json()
      })
      .then(
        (result) => {
          console.log("fetch btnFetchGetStudents= ", result);
          result.map(st => console.log(st.FullName));
          console.log('result[0].FullName=', result[0].FullName);
        },
        (error) => {
          console.log("err post=", error);
        });
```

Response will come as json

Get the data retured

If something is wrong..

- כאשר נבצע קריאה ל RESTFUL API ניתן לעשות זאת ע"י שימוש בפונקציית FETCH של הדפדפן.

```
public class StudentController : ApiController
{

    //http://localhost:49895/api/Student
    // GET: api/Student
    public IEnumerable<Student> Get()
    {
        return Student.students;
    }
```

RESTFUL API AND FETCH GET

```
fetch(this.apiUrl, {
      method: 'GET',
      headers: new Headers({
        'Content-Type': 'application/json; charset=UTF-8',
      })
    })
      .then(res => {
        console.log('res=', res);
        console.log('res.status', res.status);
        console.log('res.ok', res.ok);
        return res.json()
      })
      .then(
        (result) => {
          console.log("fetch btnFetchGetStudents= ", result);
          result.map(st => console.log(st.FullName));
          console.log('result[0].FullName=', result[0].FullName);
        },
        (error) => {
          console.log("err post=", error);
        });
```

- אותו דבר כמו בשקף הקודם אבל מפורש יותר בHEADER

```
public class StudentController : ApiController
{

    //http://localhost:49895/api/Student
    // GET: api/Student
    public IEnumerable<Student> Get()
    {
        return Student.students;
    }
```

RESTFUL API AND FETCH GET

- GET עם פרמטרים.

```
let name = 'Beyonce';
    fetch(this.apiUrl + '/' + name, {
      method: 'GET',
      headers: new Headers({
        'Content-Type': 'application/json; charset=UTF-8',
      })
    })
      .then(res => {
        console.log('res=', res);
        console.log('res.status', res.status);
        console.log('res.ok', res.ok);
        return res.json()
      })
      .then(
        (result) => {
          console.log("fetch btnFetchGetStudentByName=
          console.log('result.FullName=', result.FullName);
        },
        (error) => {
          console.log("err post=", error);
        });
```

```
//http://localhost:49895/api/student?name=beyonce and no [Route...]
//or http://localhost:49895/api/student/beyonce and the Route bellow
[Route("api/student/{name}")]
public Student Get(string name)
{
    return new Student().GetStudent(Student.students, name);
}
```

RESTFUL API AND FETCH POST

- כאשר נבצע קריאה ל RESTFUL API ניתן לעשות זאת ע"י שימוש בפונקציית FETCH של הדפדפן.
- וניתן גם לשלוח פרמטרים.

Data to send to the server

```
const s = {            //pay attention case sensitive!!!! should be exactly as the prop in C#!
    Id: "77-88",
    FullName: 'nir',
    Avg: 77.7
};

fetch(this.apiUrl, {
    method: 'POST',
    body: JSON.stringify(s),
    headers: new Headers({
      'Content-type': 'application/json; charset=UTF-8' //very important to add the 'charset=UTF-8'!!!!
    })
})
    .then(res => {
      console.log('res=', res);
      return res.json()
    })
    .then(
      (result) => {
        console.log("fetch POST= ", result);
        console.log(result.Avg);
      },
      (error) => {
        console.log("err post=", error);
      });
```

```
//POST: api/Student
//POST http://localhost:49895/api/student
//Content-type:"application/json"
//Request Body:"{ Id:77, Avg:99.9, FullName:'nir' }"
[HttpPost]
public Student Post([FromBody]Student s)
{
    Student.students.Add(s);
    return s;
}
```

RESTFUL API AND FETCH DELETE

```
fetch(this.apiUrl + '?id=5', {
      method: 'DELETE',
      //body: JSON.stringify({id:7}),
      headers: new Headers({
        'accept': 'application/json; charset=UTF-8' //very important to add the 'charset=UTF-8'!!!!
      })
    })
      .then(res => {
        console.log('res=', res);
        return res.json()
      })
      .then(
        (result) => {
          console.log("fetch POST= ", result);
        },
        (error) => {
          console.log("err post=", error);
        });
```

```
// DELETE: api/Student?id=5
[HttpDelete]
public string Delete(int id)
{
    string output = "id=" + id +"  -  ";
    output += Student.students.Count + "  -  ";
    new Student().RemoveStudent(Student.students,"303030");
    output+= Student.students.Count;
    return output;
}
```

SOME OTHER OPTIONS…

RESTFUL API AND $.AJAX GET/POST

```
var data = {
    FirstName: 'Andrew',
    LastName: 'Lock',
    Age: 31
}

$.ajax({
    type: 'POST',
    url: 'http://localhost:63597/home/PersonAsJson',
    dataType: 'json',
    contentType: 'application/x-www-form-urlencoded; charset=utf-8',
    data: data,
    success: function (result) {
        console.log("$.AJAX POST=" + result);
        console.log("$.AJAX POST=" + result.firstName);
    }
});
```

- כאשר נבצע קריאה ל RESTFUL API ניתן לעשות זאת ע"י שימוש בפונקציית $.ajax.
- יש צורך להתקין JQUERY ע"י

  npm install jquery
- import $ from 'jquery';

```
[HttpPost]
public IActionResult PersonAsJson(Person person)
{
    string str = person.FirstName;
    return Json(person);
}
```

WEB SERVICE AND $.AJAX

```
$.ajax({
    url: "http://ruppinmobile.ac.il.preview26.livedns.co.il/site01/usersws.asmx/GetAllUsers",
    dataType: "json",
    type: "POST",   //use only POST!
    data: "{}",
    contentType: "application/json; charset=utf-8",
    error: function (jqXHR, exception) {
        alert(jqXHR + exception);
    },
    success: function (data) {
        console.log(data.d);
        var users = JSON.parse(data.d);
        users.forEach(user => {
            console.log(user);
            console.log(user.Name);
        });
    }
});
```

- כאשר נבצע קריאה ל WEB ERVICE ניתן לעשות זאת ע"י שימוש בפונקציית $.ajax של JQEURY.
- יש צורך להתקין JQUERY ע"י

npm install jquery