



RESTFUL WEB SERVICE

ThS. Nguyễn Nghiệm
0913.745.789 - NghiemN@fpt.edu.vn



OBJECTIVES

- Xây dựng được Web API
- Sử dụng được Web API





AGENDA

- Ajax là gì?
- JSON là gì?
- Mô hình ứng dụng sử dụng Web API
- Xây dựng Web API
- Sử dụng Web API với jQuery
- Mô hình ứng dụng thực tế





- **AJAX (Asynchronous JavaScript and XML)**
 - ✱ Asynchronous: Mechanism
 - ✱ JavaScript: Scripting Language
 - ✱ XML: Data (TEXT, JSON)
- **Relative Concepts**
 - ✱ Rich Internet Application
 - ✱ Desktop Feeling
 - ✱ Single Page Application
- **Restful web service**



The diagram illustrates the interaction between a Browser Client and a Web Server over time. The timeline shows alternating periods of activity (requests and responses) and inactivity (idle time) for both parties.

- Browser Client:** Represented by blue hatched rectangles at the top.
- Web Server:** Represented by yellow hatched rectangles at the bottom.
- Time:** Indicated by a horizontal axis at the bottom.
- Sequence of Events:**
 - The Browser Client sends a request (blue solid arrow) to the Web Server.
 - The Web Server responds (green dotted arrow) to the Browser Client.
 - This cycle repeats three times, showing alternating periods of activity and inactivity.

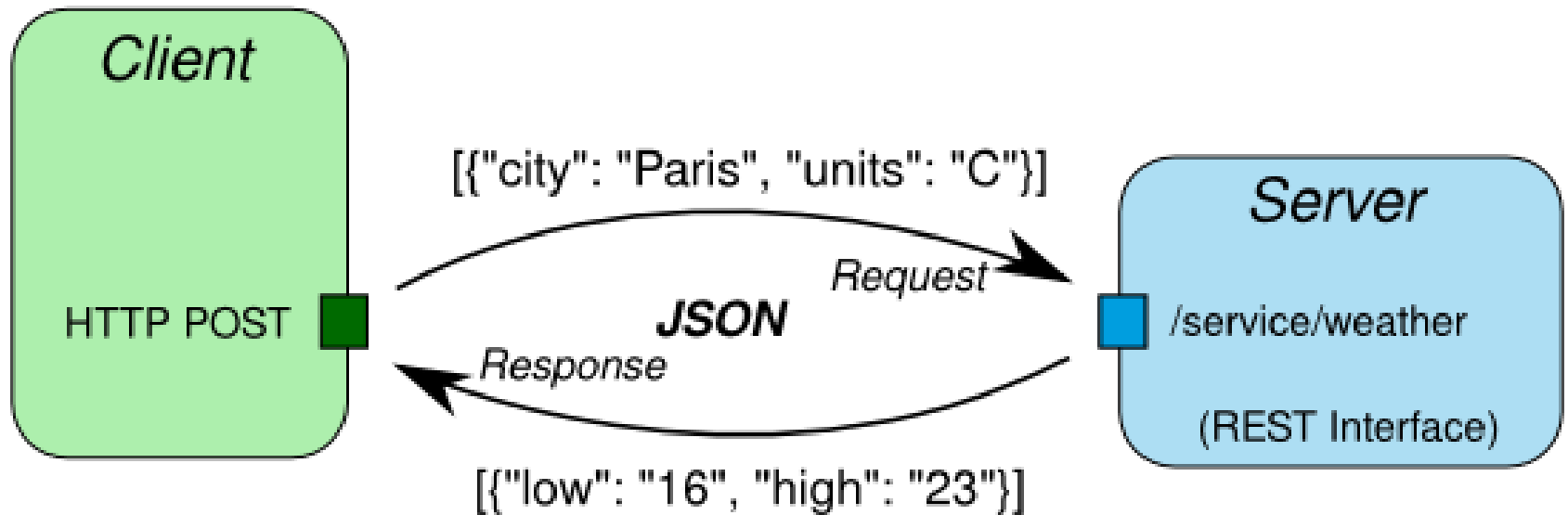
The diagram shows a sequence of requests and responses between a Browser Client and a Web Server over time. The Browser Client is represented by a blue box at the top, and the Web Server is represented by a series of yellow boxes at the bottom. Solid blue arrows represent requests from the client to the server, and dotted green arrows represent responses from the server back to the client. The horizontal axis is labeled 'Time'.





JSON COMMUNICATION

JSON / REST / HTTP





JSON DATA

- Data

- ✴ Object: {prop1: value1, prop2:value2,...}

- ✴ Array of Objects: [{...}, {...},...]

- Chuyển đổi JSON và String

- ✴ JSON.stringify(json):string

- ✴ JSON.parse(string):json

- Ví dụ

- ✴ var data = {"hoten":"Nguyễn Văn Tèo", tuoi:20};

- ✴ alert(data.hoten)

- ✴ alert(JSON.stringify(data));



AJAX AND WEB API

```
$.ajax({  
  url:'api-url',  
  data: json,  
  type:'GET|POST|PUT|DELETE',  
  contentType:'application/json'  
  success: function(response){  
    // success handling code  
  },  
  error: function(){  
    // error handling code  
  }  
})
```

Request

ActionMethod

Response

@RestController



SIMPLE EXAMPLE

```
@RestController
@RequestMapping(value="api/hello",
    produces=MediaType.APPLICATION_JSON_UTF8_VALUE)
public class HelloController {
    @GetMapping()
    public ResponseEntity<String[]> get(){
        String[] data = {"Tuấn", "Phương", "Hằng"};
        return ResponseEntity.ok(data);
    }
}
```

GET: api/hello

```
$.ajax({
    url: "api/hello",
    type: "GET",
    success: function (response) {
    }
});
```

["Tuấn", "Phương", "Hằng"]



WEB API

```
@RestController
@RequestMapping(value="api/categories",
    produces=MediaType.APPLICATION_JSON_UTF8_VALUE)
public class ApiController {
    @Autowired
    CategoryDAO dao;

    @GetMapping() //GET:api/categories
    public ResponseEntity<List<Category>> get(){...}

    @GetMapping("/{id}") //GET:api/categories/{id}
    public ResponseEntity<Category> get(@PathVariable("id") Integer id){...}

    @PostMapping() //POST:api/categories
    public ResponseEntity<Category> post(@RequestBody Category category){...}

    @PutMapping("/{id}") //PUT:api/categories/{id}
    public ResponseEntity<Void> put(
        @PathVariable("id") Integer id, @RequestBody Category category){...}

    @DeleteMapping("/{id}") //DELETE:api/categories/{id}
    public ResponseEntity<Void> delete(@PathVariable("id") Integer id){...}
}
```



WEB API - GET

```
@GetMapping("api/categories")  
public ResponseEntity<List<Category>> get(){  
    return ResponseEntity.ok(dao.findAll());  
}
```

```
$.ajax({  
    url: "api/categories",  
    type: "GET",  
    success: function (response) {  
        alert("Tải danh sách loại thành công")  
    },  
    error: function () {  
        alert("Lỗi")  
    }  
});
```

GET:api/categories

Get()

[{id:?, name:?, nameVN:?},...]



WEB API - GET

```
@GetMapping("api/categories/{id}")  
public ResponseEntity<Category> get(@PathVariable("id") Integer id){  
    return ResponseEntity.ok(dao.findById(id));  
}
```

```
$.ajax({  
    url: "api/categories/1236",  
    type: "GET",  
    success: function (response) {  
        alert("Tải một loại thành công")  
    },  
    error: function () {  
        alert("Lỗi")  
    }  
});
```

GET:api/categories/id

{id:?, name:?, nameVN:??}

Get(id)



WEB API - POST

```
@PostMapping("api/categories")
public ResponseEntity<Category> post(@RequestBody Category category){
    dao.create(category);
    return ResponseEntity.ok(category);
}
```

```
$ajax({
  url: "api/categories",
  data: JSON.stringify({
    name: "Rose",
    nameVN: "Hoa hồng"
  }),
  contentType: 'application/json',
  type: "POST",
  success: function (response) {
    alert("Thêm mới thành công")
  },
  error: function () {
    alert("Lỗi")
  }
});
```

POST:api/categories
{name:?, nameVN:??}

{id:?, name:?, nameVN:??}

Post(data)



WEB API - PUT

```
@PutMapping("api/categories/{id}")
public ResponseEntity<Void> put(
    @PathVariable("id") Integer id, @RequestBody Category category){
    if(dao.exist(id)) {
        dao.update(category);
        return ResponseEntity.ok().build();
    }
    return ResponseEntity.notFound().build();
}
```

```
$.ajax({
    url: "api/categories/1236",
    data: JSON.stringify({
        id:1236,
        name: "Rose",
        nameVN: "Hoa hồng"
    }),
    contentType: 'application/json',
    type: "PUT",
    success: function (response) {
        alert("Cập nhật thành công")
    },
    error: function () {
        alert("Lỗi")
    }
});
```

PUT:api/categories/id
{id:?, name:?, nameVN:??}

Put(id, data)

}



WEB API - DELETE

```
@DeleteMapping("api/categories/{id}")
public ResponseEntity<Void> delete(@PathVariable("id") Integer id){
    if(dao.exist(id)) {
        dao.delete(id);
        return ResponseEntity.ok().build();
    }
    return ResponseEntity.notFound().build();
}
```

```
$.ajax({
    url: "api/categories/1236",
    type: "DELETE",
    success: function (response) {
        alert("Xóa thành công")
    },
    error: function () {
        alert("Lỗi")
    }
});
```

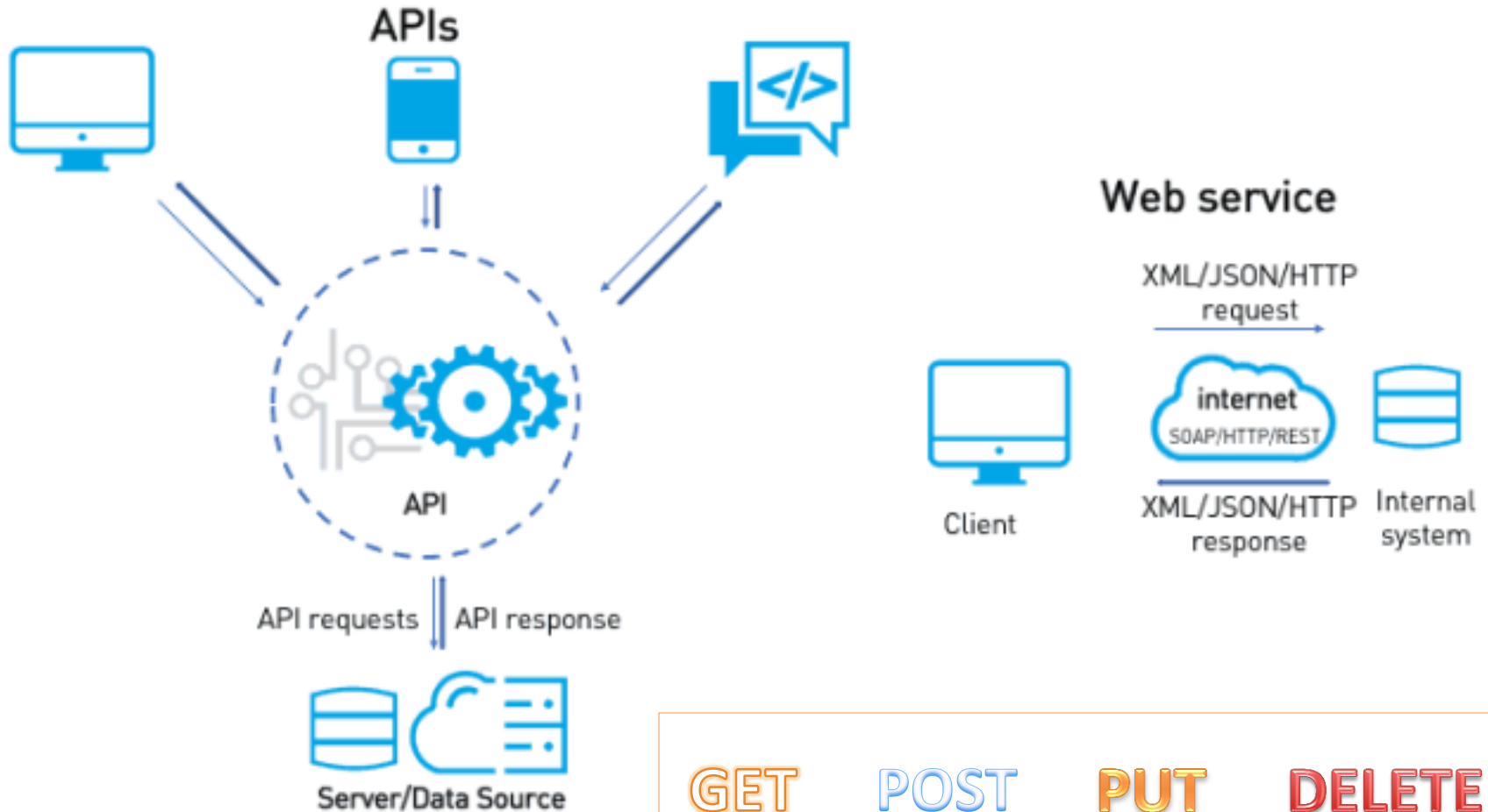
DELETE:api/categories/id

Delete(id)

}



RESTFUL WEB SERVICE





SUMMARY

- Ajax là gì?
- JSON là gì?
- Mô hình ứng dụng sử dụng Web API
- Xây dựng Web API
- Sử dụng Web API với jQuery
- Mô hình ứng dụng thực tế

