REST API Security

Unexpected behavior? Expect bugs!

ocado TECHNOLOGY

whoami

- Kuba Maćkowski (LI, jakub.mackowski@ocado.com)
- AppSec Engineer
  - in Ocado from January 2018
- Before ~6 years as a: dev, tester, qa -> migrating towards application security

what is not coverd in this talk (but is still important!)

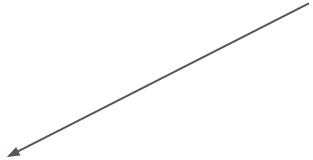
- “Standart” Web app security (SQLi, XXE, ...)
- API key leaks
- authentication/authorization
- Transport Security, MiTM, HTTPS (use TLS!)

agenda

```
POST /greeting?id=1 HTTP/1.1
Host: localhost:8080
Content-Type: application/json
Accept: application/json
Content-Length: 24

{"name":"Ocado", "id":0}
```

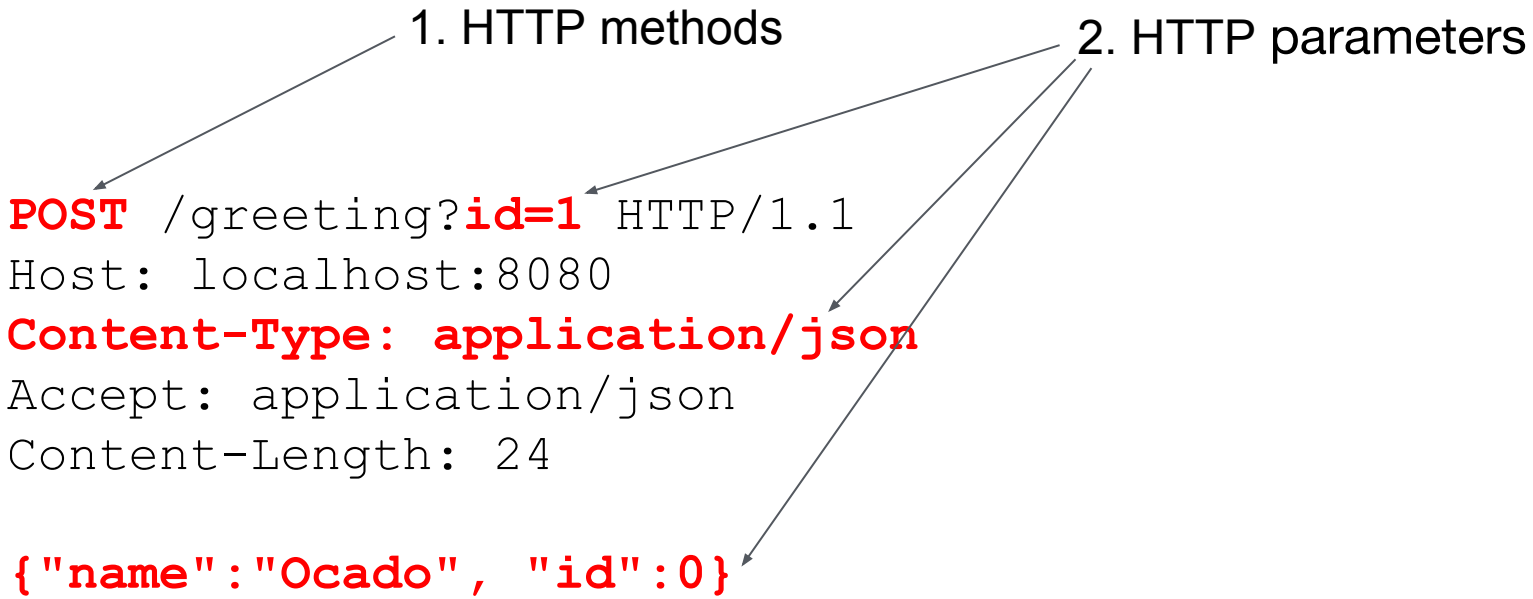
agenda

1. HTTP methods

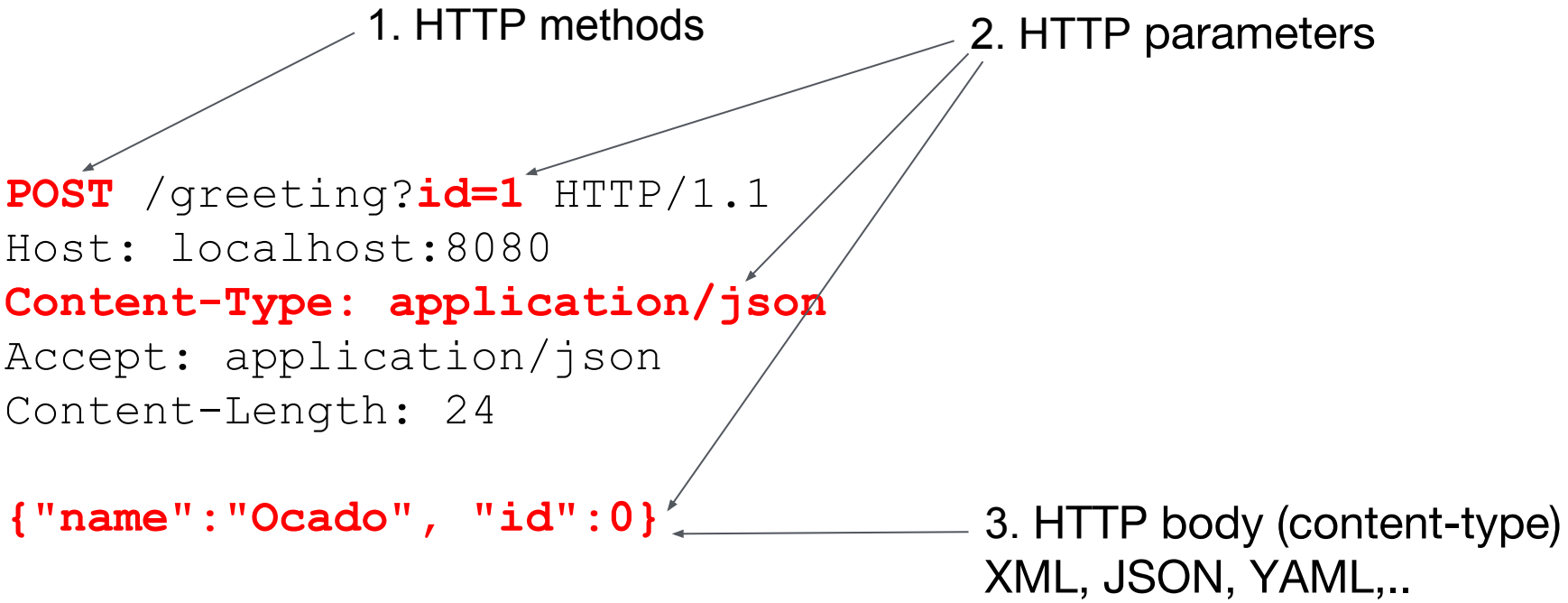


```
POST /greeting?id=1 HTTP/1.1
Host: localhost:8080
Content-Type: application/json
Accept: application/json
Content-Length: 24

{"name":"Ocado", "id":0}
```

agenda


1. HTTP methods
2. HTTP parameters
POST /greeting?id=1 HTTP/1.1
Host: localhost:8080
Content-Type: application/json
Accept: application/json
Content-Length: 24
{"name":"Ocado", "id":0}

agenda


1. HTTP methods
2. HTTP parameters
POST /greeting?id=1 HTTP/1.1
Host: localhost:8080
Content-Type: application/json
Accept: application/json
Content-Length: 24
{"name":"Ocado", "id":0}
3. HTTP body (content-type)
XML, JSON, YAML,..

Getting Started · Building a RESTful Web Service - Spring

https://spring.io/guides/gs/rest-service/ “spring boot rest api example”

```
@RestController
public class GreetingController {

  private static final String template = "Hello, %s!";
  private final AtomicLong counter = new AtomicLong();

  @RequestMapping("/greeting")
  public Greeting greeting( @RequestParam(value="name") String name) {
    return new Greeting(counter.incrementAndGet(), String.format(template, name));
  }

}
```

HTTP methods

- GET /greeting
- POST /greeting
- DELETE /greeting
- OPTIONS - list of methods
- TRACE - method not allowed
- MERGE - but this one is allowed
- OCADO - and this one also :-)

HTTP methods

- What is your framework doing for you?
  - Full CRUD for entity class
  - Auto-generated HEAD for GET resources
  - Support for OPTIONS and TRACE methods
  - Docs
- Why this can be dangerous?
  - Forced browsing (DELETE)
  - Information leakage (TRACE, OPTIONS, error)
  - **Authentication\Authorization bypass (if access control is implemented based on HTTP method)**
  - Firewall (WAF, AWS WAF) rules bypass

HTTP methods

```
<http auto-config="true">
 <intercept-url pattern="/api/" access="isAuthenticated" method="GET" />
 <intercept-url pattern="/api/" access="hasRole('EDITOR')" method="POST" />
</http>
```

Java annotation-based configuration

```
http.authorizeRequests().antMatchers(HttpMethod.GET).permitAll();
http.authorizeRequests().antMatchers(HttpMethod.POST).denyAll();
http.authorizeRequests().antMatchers(HttpMethod.DELETE,"/you/can/alsoSpecif
yAPath").denyAll();
```

HTTP method override - why?

“

I got an email today where someone had built a REST(ful/ish) API with ASP.NET Web API that had a customer who was against the idea of using GET, POST, PUT, and DELETE, and insisted that they only use GET and POST.

Sometimes this is because of a **browser or client limitaton**, sometimes it's a really **tense corporate firewall**. They wanted to know what they could do.

“

Using the PUT method, you can upload any file on the server
TRACE, OPTIONS - diagnostic/debug methods

HTTP method override - why?

“

One thing you can do is to "tunnel" HTTP Methods inside another HTTP Header. Basically you have a header that says "No, seriously, I know I got here via a POST, but use this one instead." You would still POST, but then you'd have "**X-HTTP-Method-Override:PUT**" as a header.

“

https://www.hanselman.com/blog/HTTPPUTOrDELETENotAllowedUseXHTTPMethodOverrideForYourRESTServiceWithASPNETWebAPI.aspx

HTTP method override - in WP

Some servers and clients cannot correctly process some HTTP methods that the API makes use of. For example, all deletion requests on resources use the DELETE method, but some clients do not provide the ability to send this method.

To ensure compatibility with these servers and clients, the API supports a method override. This can be passed either via a **_method parameter** or the **X-HTTP-Method-Override header**, with the value set to the HTTP method to use.

https://developer.wordpress.org/rest-api/using-the-rest-api/global-parameters/

HTTP method override - in nodeJS

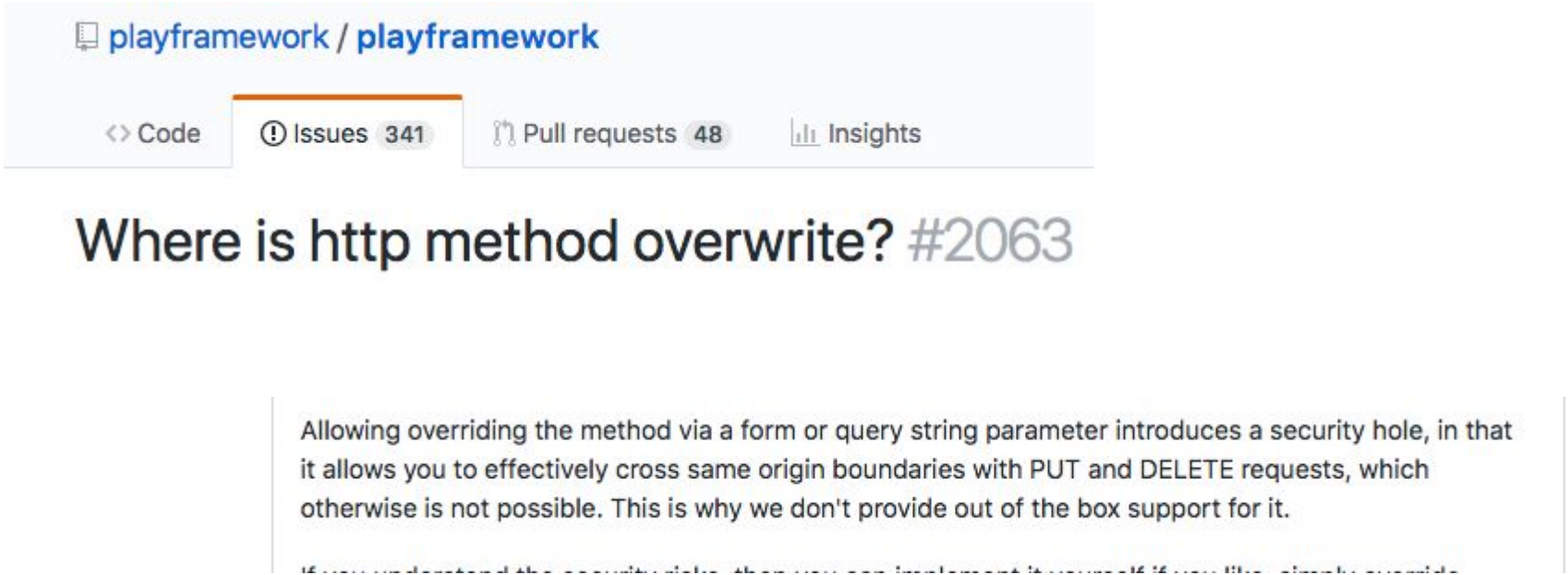
```
var express = require('express')
var methodOverride = require('method-override')
var app = express()

// override with different headers; last one takes precedence
app.use(methodOverride('X-HTTP-Method')) //          Microsoft
app.use(methodOverride('X-HTTP-Method-Override')) // Google/GData
app.use(methodOverride('X-Method-Override')) //      IBM
```

https://github.com/expressjs/method-override

HTTP method override

- It is often **enabled by default**
  (in most .net frameworks,
  php frameworks, symfony, CakePHP,
  java\scala - play framework 1.X,
  Spring - HiddenHttpMethodFilter.class)
- Sometimes it can be enabled:
  - Nodejs: npm install method-override

but there is hope!



https://github.com/playframework/playframework/issues/2063

method override, is it a problem?

- Authentication bypass
  e.g. When you allow GET but not DELETE and method override is done after permissions check
- Firewall (WAF, AWS WAF) rules bypass (these are not aware of method override)
- CSRF Token Bypass
- Make easier to exploit
  - Cross site request forgery
  - Server site request forgery

Cisco Prime Infrastructure case study

- https://www.security-assessment.com/files/documents/advisory/Cisco-Prime-Infrastructure-Release.pdf

HTTP parameters

Where i can find parameters in HTTP request?

- URL
  - “directory” e.g. /api/users/3/email
  - “standard” e.g. /api/users/email?id=3
- HTTP headers
- HTTP body
- Mix of all

HTTP parameters pollution

```
POST /greeting?name=Ocado HTTP/1.1
Host: localhost:8080
Content-Length: 1

HTTP/1.1 200
Content-Type: application/json;charset=UTF-8
Date: Fri, 13 Jul 2018 09:09:35 GMT
Content-Length: 35

{"id":1,"content":"Hello, Ocado!"}
```

HTTP parameters pollution

Do you know what will happen when:

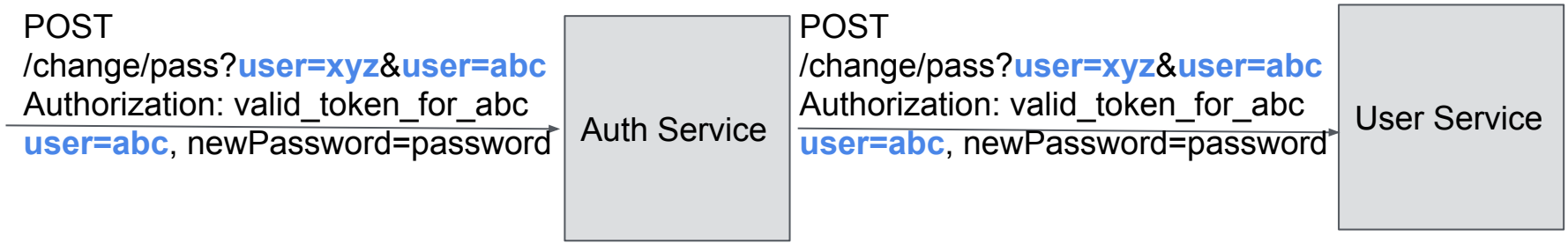
```
POST /greeting?name=Barcelona

POST /greeting?name=Barcelona&name=Cracow

POST /greeting?name=Barcelona&name=Cracow
Content-Type: application/x-www-form-urlencoded
name=Sofia
```

Ok cool ... but what does this have to do with security?

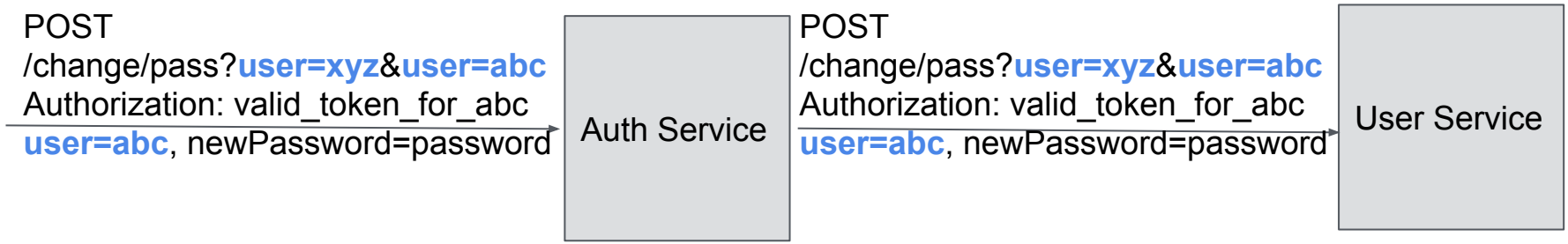
```
if(checkAccess(request)) {
    return doSomething(request.getRequestBody().getName())
} else {
  return 401 Unauthorized HTTP responses
}
...
checkAccess(request) {
  User = request.getRequestParam().getUser()
  if user.hasAccessTo(request.getRequestParam().getName()) {
    return true
  } else {
    return false
  }
}
```

ocado TECHNOLOGY

Ok cool ... but what does this have to do with security?

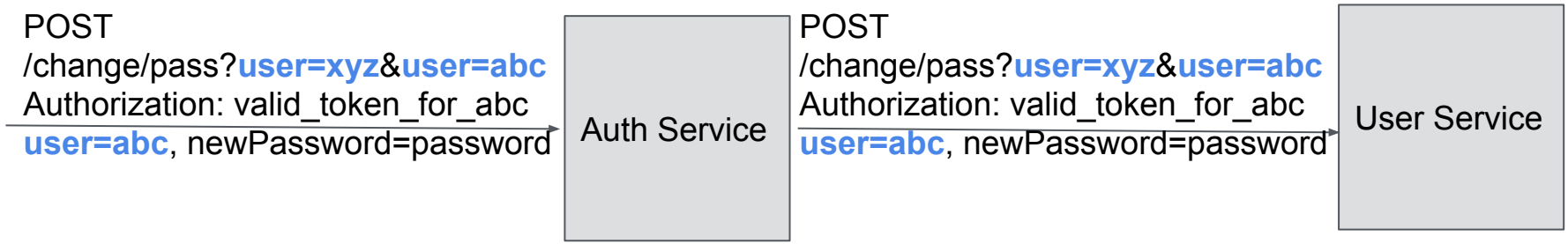


Ok cool ... but what does this have to do with security?



- Not all frameworks works in the same way!!!
  - String concatenation
  - First, last, error, [ ]

Ok cool ... but what does this have to do with security?



- Cases from the wild? https://blog.sucuri.net/2017/02/content-injection-vulnerability-wordpress-rest-api.html

Content Injection Vulnerability in WordPress REST API (2017)

- “One of these REST endpoints allows access (via the API) to view, edit, delete and create posts. **Within this particular endpoint, a subtle bug allows visitors to edit any post on the site.**”
- **/wp-json/wp/v2/posts/1234** – the ID parameter would be set to 1234
- **/wp-json/wp/v2/posts/1234?id=1234helloworld** – which would assign 1234helloworld to the ID parameter – which now contains more than just digits.
- WP validates the request by checking if the post actually exists and whether our user has permission to edit this post
- WordPress casts the ID parameter to an integer before passing it to get_post

HTTP body (content-type)

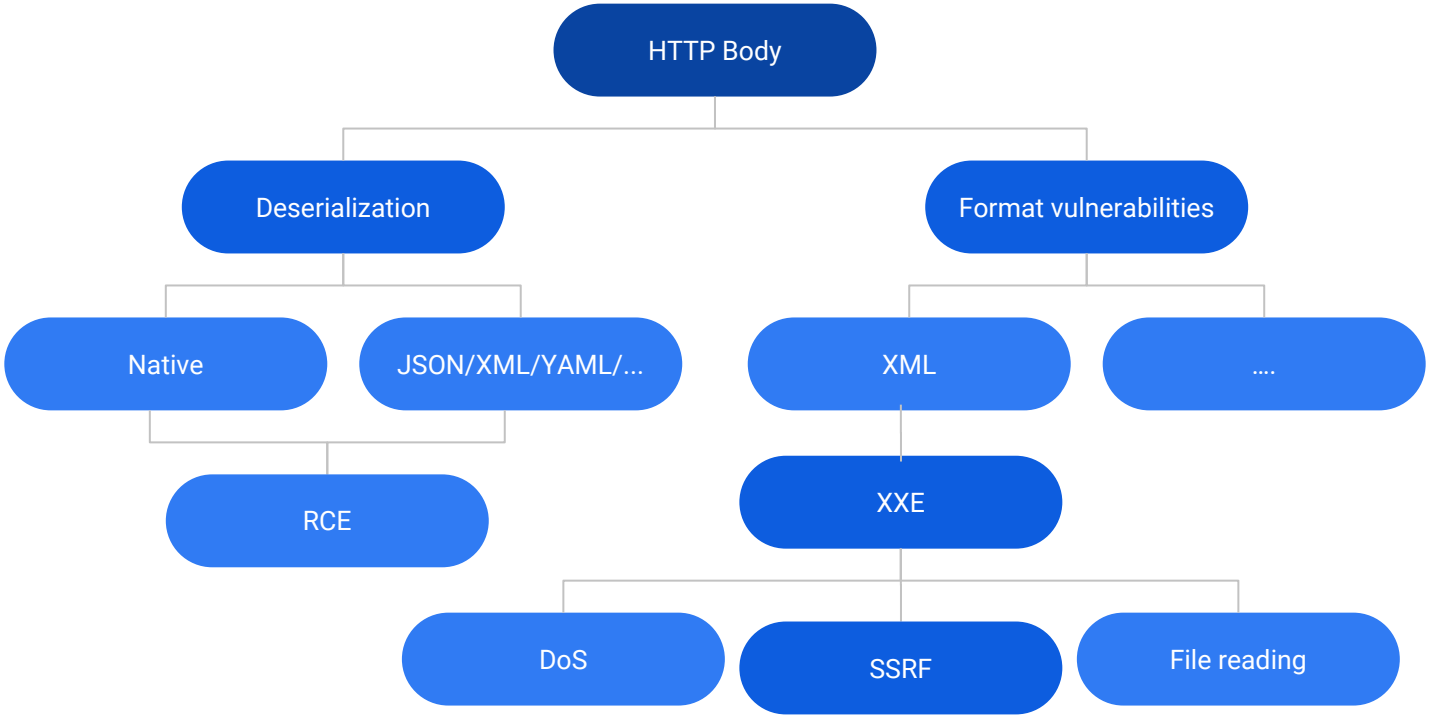
- Why Spring boot return json?
- Lets add Accept: application/json
- Change it to application/xml -> http 406 not acceptable!
- Cool! I am using JSON API :-)
- No not really :-)

HTTP body (content-type)

```
function rest_server_request_parsers() {
    static $parsers = NULL;
    if (!$parsers) {
        $parsers = array(
            'application/x-www-form-urlencoded' => 'ServicesParserURLEncoded',
            'application/json' => 'ServicesParserJSON',
            'application/vnd.php.serialized' => 'ServicesParserPHP',
            'multipart/form-data' => 'ServicesParserMultipart',
            'application/xml' => 'ServicesParserXML',
            'text/xml' => 'ServicesParserXML',
        );
    }
}
```

One more time, cool trick...

- … but why/when this is dangerous?



key takeaways

- Use the **proper HTTP method** according to the operation: GET (read), POST (create), PUT/PATCH (replace/update), and DELETE (to delete a record), and respond with **405 Method Not Allowed** if the requested method isn't appropriate for the requested resource.

key takeaways

- **Validate content-type on request Accept header** (Content Negotiation) to allow only your supported format (e.g. application/xml, application/json, etc) and respond with **406 Not Acceptable** response if not matched.

key takeaways

- **Validate content-type of posted data** as you accept (e.g. application/x-www-form-urlencoded, multipart/form-data, application/json, etc).

key takeaways

- **Force content-type for your response**, if you return application/json then your response content-type is application/json.

key takeaways

- Find out what your framework is doing :-)
  - Disable HTTP method override
  - Check how multiple parameters are handled
- https://github.com/GrrrDog/Java-Deserialization-Cheat-Sheet
- https://github.com/shieldfy/API-Security-Checklist

Questions?

Did you learn something new today?

/OcadoTechnology

ocado TECHNOLOGY