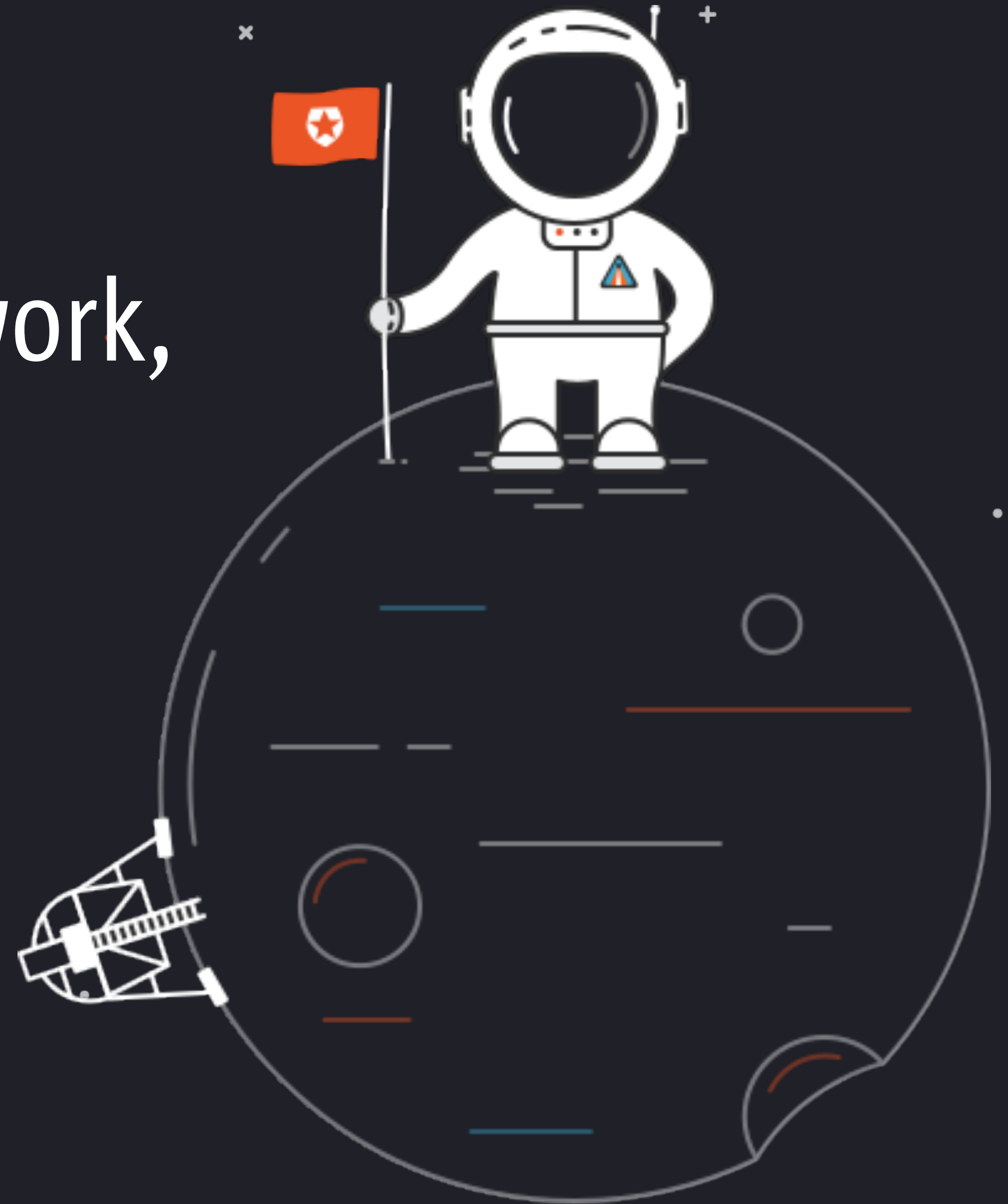


Rest Services with Play Framework, and a Security Level with JWT (JSON Web Tokens)

ODT LATAM





Mercedes Wyss
@itrjwyss



Community Leader
JDuchess Chapter Guatemala

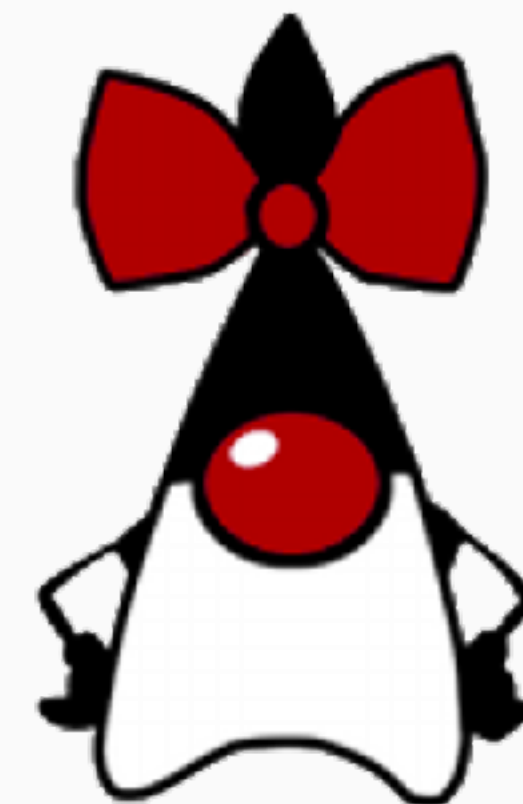
Ex-JUG Member
Guatemala Java Users Group (GuateJUG)

Chief Technology Officer (CTO) at Produactivity
Full Stack Developer

Auth0 Ambassador



Auth0



Agenda

- Play Framework
- Web Services Restful
- JSON
- JWT (JSON Web Tokens)



Agenda

- **Play Framework**
- Web Services Restful
- JSON
- JWT (JSON Web Tokens)



xplay

Play Framework^x

- Scala Based
- Lightweight^x, stateless, web-friendly
- Minimal resources consumption for highly-scalable applications



CloudWatch Monitoring Details

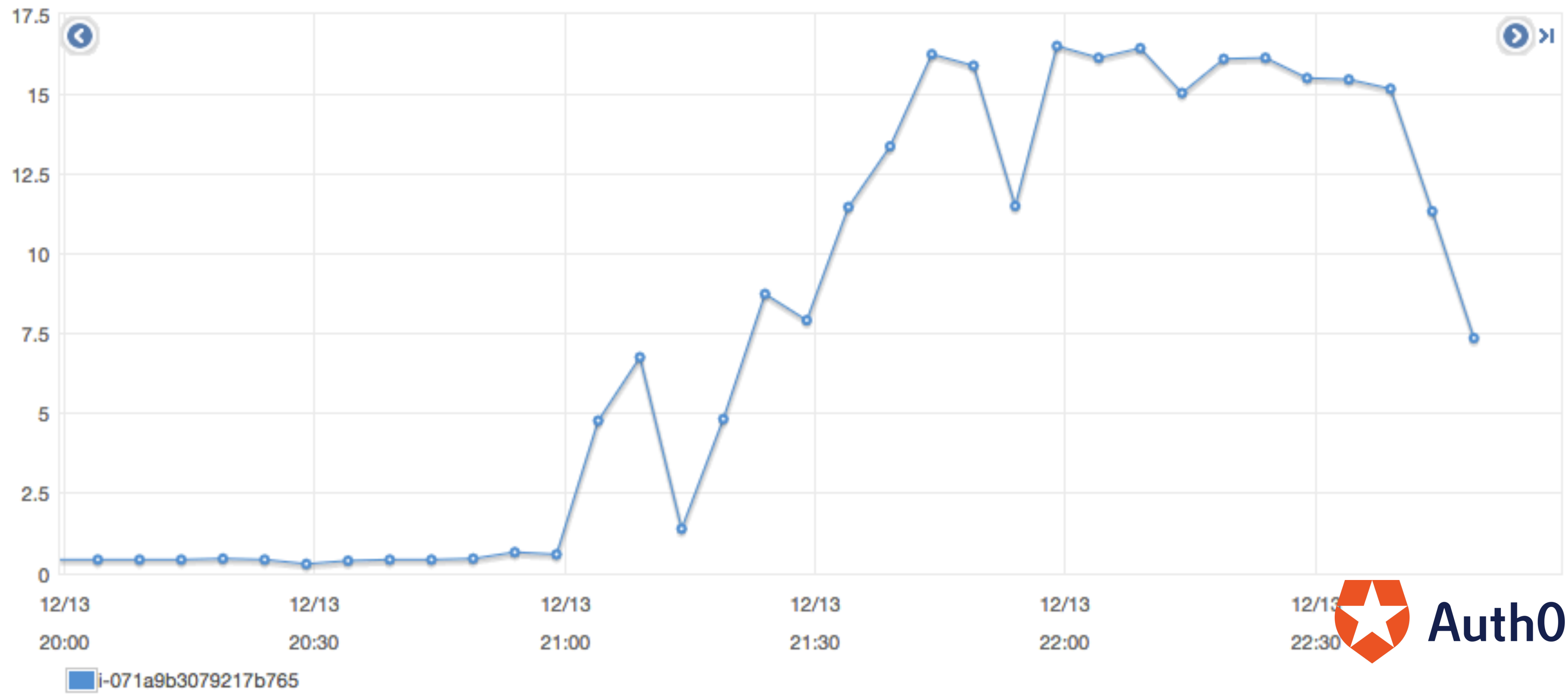


CPU Utilization (Percent)

Statistic: Average ▾

Time Range: Last 3 Hours ▾

Period: 5 Minutes ▾



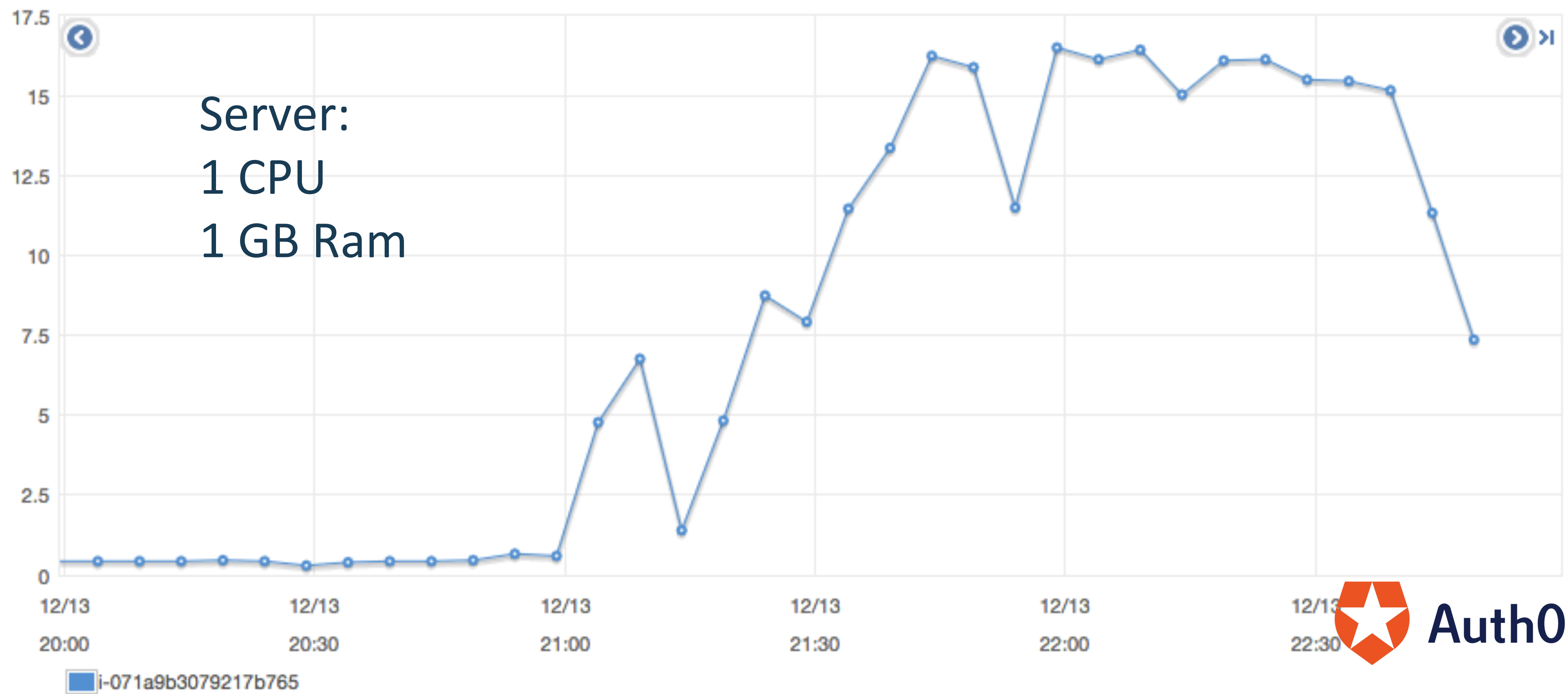
CPU Utilization (Percent)

Statistic: Average ▾

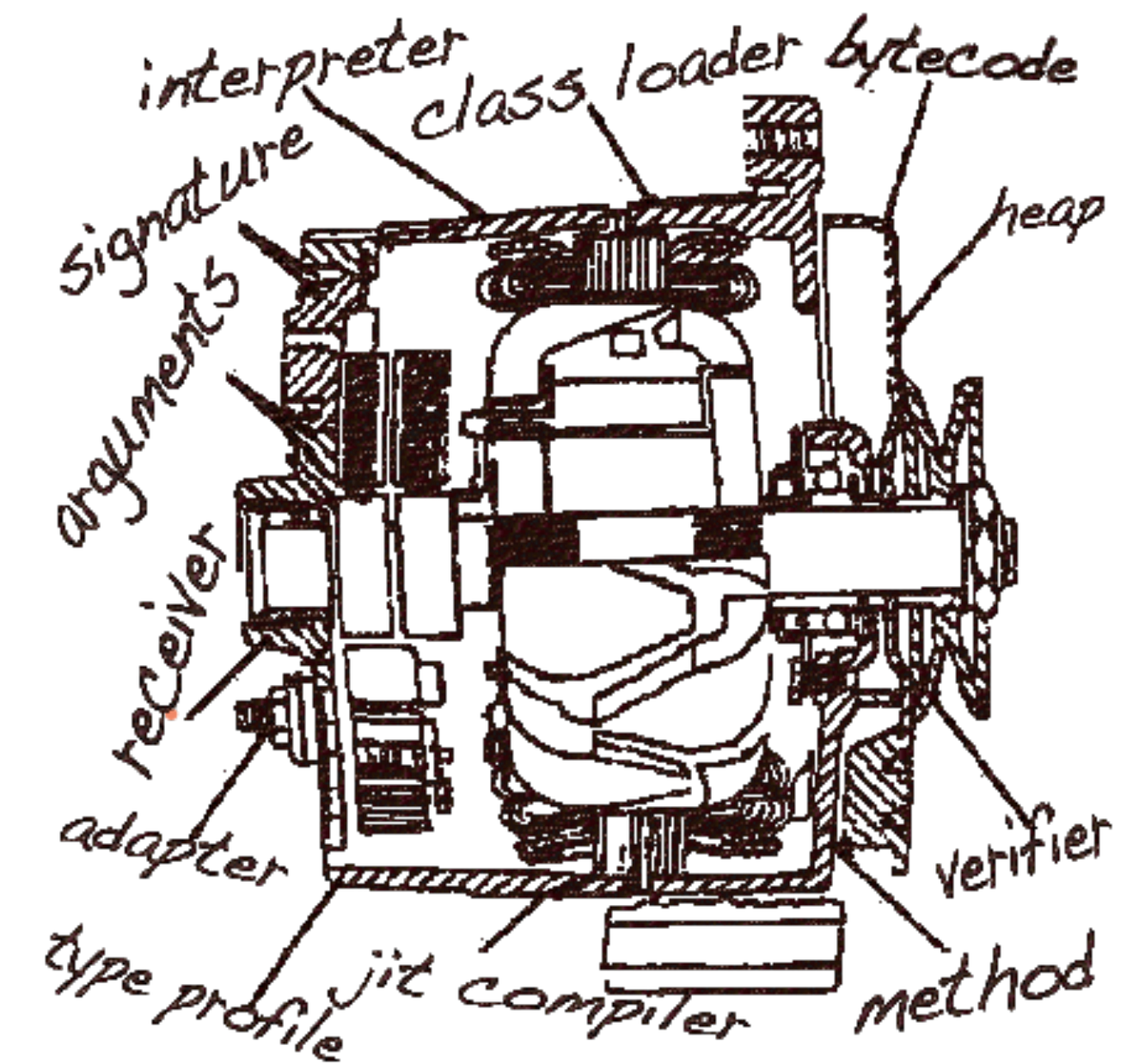
Time Range: Last 3 Hours ▾

Period: 5 Minutes ▾

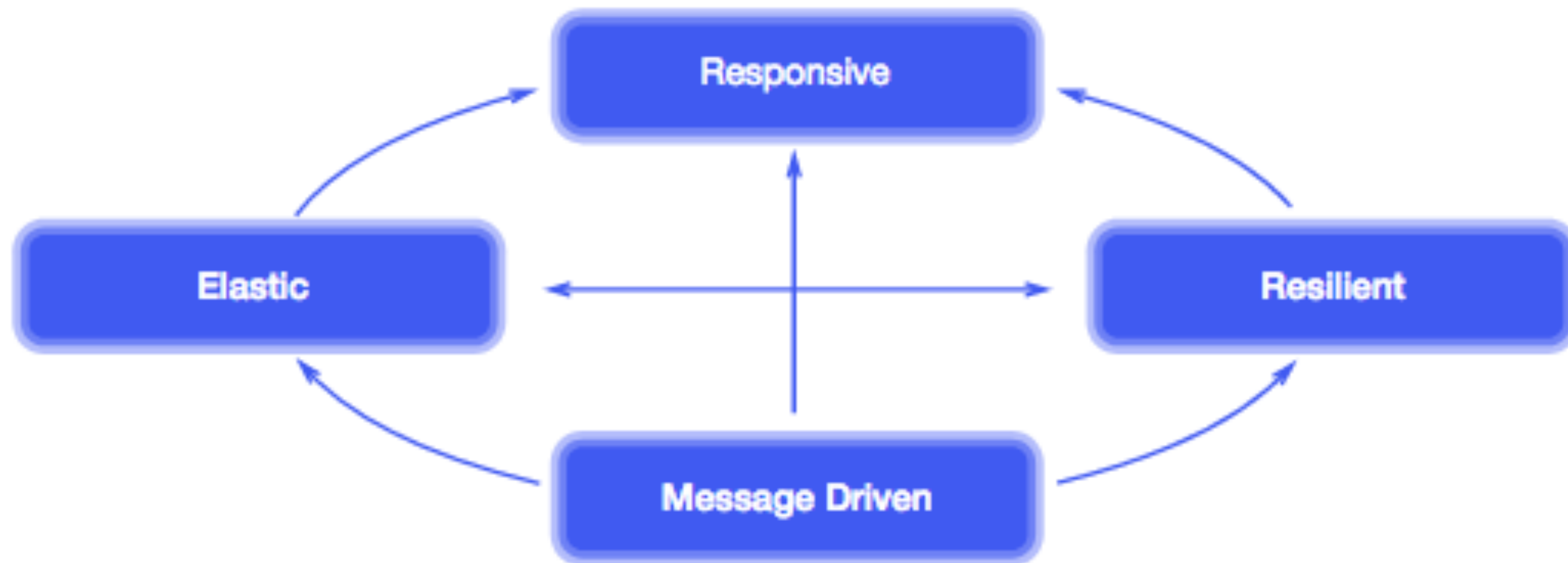
↺↻



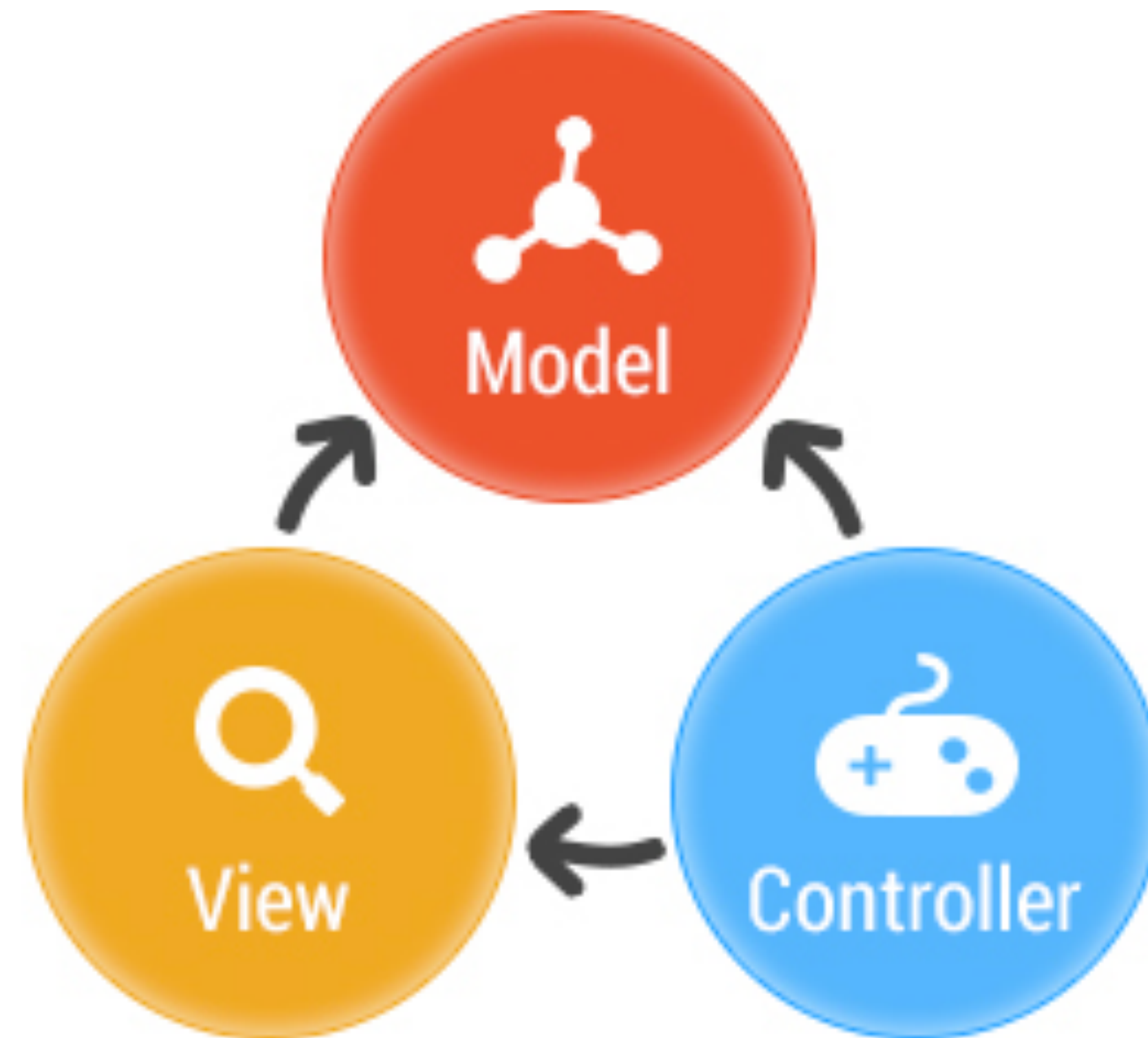
Java or Scala



Is Reactive



MVC Architecture



App Server



+ Since Version 2.6.x

+ Fewer Configurations



JRebel Behavior

More Code, Less Deploys



• How start a Play project?

- We need **SBT**

- **Create**

sbt new playframework/play-java-seed.g8

• sbt new playframework/play-scala-seed.g8

- **Deploy**

sbt run



```
$ sbt new playframework/play-java-seed.g8
```

This template generates a Play Java project

```
name [play-java-seed]: ODT
organization [com.example]: com.auth0
scala_version [2.12.3]:
play_version [2.6.7]:
sbt_version [1.0.2]:
```

Template applied in ./odt

```
ODT $ sbt run
```




```
[info] Loading project definition from /Users/mercedeswyss/Documents/Presentaciones/2017/OracleCode/Mexico/play-java-starter-example/project
[info] Updating {file:/Users/mercedeswyss/Documents/Presentaciones/2017/OracleCode/Mexico/play-java-starter-example/project/}play-java-starter-example-build...
[info] Resolving org.fusesource.jansi#jansi;1.4 ...
[info] downloading https://repo.scala-sbt.org/scalasbt/sbt-plugin-releases/com.typesafe.play/sbt-plugin/scala_2.10/sbt_0.13/2.6.0/jars/sbt-plugin.jar ...
[info] [SUCCESSFUL] com.typesafe.play#sbt-plugin;2.6.0!sbt-plugin.jar (3794ms)
[info] downloading https://repo.scala-sbt.org/scalasbt/sbt-plugin-releases/com.typesafe.sbt/sbt-play-enhancer/scala_2.10/sbt_0.13/1.1.0/jars/sbt-play-enhancer.jar ...
[info] [SUCCESSFUL] com.typesafe.sbt#sbt-play-enhancer;1.1.0!sbt-play-enhancer.jar (2439ms)
[info] downloading https://repo1.maven.org/maven2/com/typesafe/play/sbt-routes-compiler_2.10/2.6.0/sbt-routes-compiler_2.10-2.6.0.jar ...
[info] [SUCCESSFUL] com.typesafe.play#sbt-routes-compiler_2.10;2.6.0!sbt-routes-compiler_2.10.jar (961ms)
[info] downloading https://repo1.maven.org/maven2/com/typesafe/play/run-support_2.10/2.6.0/run-support_2.10-2.6.0.jar ...
[info] [SUCCESSFUL] com.typesafe.play#run-support_2.10;2.6.0!run-support_2.10.jar (441ms)
[info] downloading https://repo1.maven.org/maven2/com/typesafe/config/1.3.1/config-1.3.1.jar ...
[info] [SUCCESSFUL] com.typesafe#config;1.3.1!config.jar(bundle) (494ms)
[info] downloading https://repo.scala-sbt.org/scalasbt/sbt-plugin-releases/net.contentobjects.jnotify/jnotify/0.94-play-1/jars/jnotify.jar ...
[info] [SUCCESSFUL] net.contentobjects.jnotify#jnotify;0.94-play-1!jnotify.jar (8790ms)
[info] downloading https://repo.scala-sbt.org/scalasbt/sbt-plugin-releases/com.typesafe.sbt/sbt-twirl/scala_2.10/sbt_0.13/1.3.2/jars/sbt-twirl.jar ...
[info] [SUCCESSFUL] com.typesafe.sbt#sbt-twirl;1.3.2!sbt-twirl.jar (2286ms)
[info] downloading https://repo.scala-sbt.org/scalasbt/sbt-plugin-releases/com.typesafe.sbt/sbt-native-packager/scala_2.10/sbt_0.13/1.2.0/jars/sbt-native-packager.jar ...
[info] [SUCCESSFUL] com.typesafe.sbt#sbt-native-packager;1.2.0!sbt-native-packager.jar (4539ms)
[info] downloading https://repo.scala-sbt.org/scalasbt/sbt-plugin-releases/com.lightbend.sbt/sbt-javaagent/scala_2.10/sbt_0.13/0.1.2/jars/sbt-javaagent.jar ...
[info] [SUCCESSFUL] com.lightbend.sbt#sbt-javaagent;0.1.2!sbt-javaagent.jar (2553ms)
[info] downloading https://repo.scala-sbt.org/scalasbt/sbt-plugin-releases/com.typesafe.sbt/sbt-web/scala_2.10/sbt_0.13/1.4.1/jars/sbt-web.jar ...
[info] [SUCCESSFUL] com.typesafe.sbt#sbt-web;1.4.1!sbt-web.jar (2814ms)
[info] downloading https://repo.scala-sbt.org/scalasbt/sbt-plugin-releases/com.typesafe.sbt/sbt-js-engine/scala_2.10/sbt_0.13/1.2.1/jars/sbt-js-engine.jar ...
[info] [SUCCESSFUL] com.typesafe.sbt#sbt-js-engine;1.2.1!sbt-js-engine.jar (2613ms)
[info] downloading https://repo1.maven.org/maven2/com/typesafe/play/twirl-api_2.10/1.3.2/twirl-api_2.10-1.3.2.jar ...
[info] [SUCCESSFUL] com.typesafe.play#twirl-api_2.10;1.3.2!twirl-api_2.10.jar (1042ms)
[info] downloading https://repo1.maven.org/maven2/commons-io/commons-io/2.5/commons-io-2.5.jar ...
[info] [SUCCESSFUL] commons-io#commons-io;2.5!commons-io.jar (949ms)
[info] downloading https://repo1.maven.org/maven2/com/typesafe/play/build-link/2.6.0/build-link-2.6.0.jar ...
[info] [SUCCESSFUL] com.typesafe.play#build-link;2.6.0!build-link.jar (513ms)
[info] downloading https://repo1.maven.org/maven2/com/lightbend/play/play-file-watch_2.10/1.0.0/play-file-watch_2.10-1.0.0.jar ...
[info] [SUCCESSFUL] com.lightbend.play#play-file-watch_2.10;1.0.0!play-file-watch_2.10.jar (563ms)
[info] downloading https://repo1.maven.org/maven2/com/typesafe/play/play-exceptions/2.6.0/play-exceptions-2.6.0.jar ...
[info] [SUCCESSFUL] com.typesafe.play#play-exceptions;2.6.0!play-exceptions.jar (523ms)
[info] downloading https://repo1.maven.org/maven2/com/lightbend/play/jnotify/0.94-play-2/jnotify-0.94-play-2.jar ...
[info] [SUCCESSFUL] com.lightbend.play#jnotify;0.94-play-2!jnotify.jar (2244ms)
[info] downloading https://repo1.maven.org/maven2/com/github/pathikrit/better-files_2.10/2.14.0/better-files_2.10-2.14.0.jar ...
[info] [SUCCESSFUL] com.github.pathikrit#better-files_2.10;2.14.0!better-files_2.10.jar (1089ms)
[info] downloading https://repo1.maven.org/maven2/com/typesafe/play/twirl-compiler_2.10/1.3.2/twirl-compiler_2.10-1.3.2.jar ...
[info] [SUCCESSFUL] com.typesafe.play#twirl-compiler_2.10;1.3.2!twirl-compiler_2.10.jar (1010ms)
[info] downloading https://repo.scala-sbt.org/scalasbt/sbt-plugin-releases/org.scala-js/sbt-scalajs/scala_2.10/sbt_0.13/0.6.16/jars/sbt-scalajs.jar ...
[info] [SUCCESSFUL] org.scala-js#sbt-scalajs;0.6.16!sbt-scalajs.jar (5827ms)
[info] downloading https://repo1.maven.org/maven2/com/typesafe/play/twirl-parser_2.10/1.3.2/twirl-parser_2.10-1.3.2.jar ...
```



Auth0


```
[warn] Executing in batch mode.  
[warn]   For better performance, hit [ENTER] to switch to interactive mode, or  
[warn]   consider launching sbt without any commands, or explicitly passing 'shell'  
[info] Loading project definition from /Users/mercedeswyss/Documents/Presentaciones/2017/OracleCode/Sydney/OracleCodeSydney/project  
[info] Set current project to OracleCodeSydney (in build file:/Users/mercedeswyss/Documents/Presentaciones/2017/OracleCode/Sydney/OracleCodeSydney/)
```

--- (Running the application, auto-reloading is enabled) ---

```
[info] p.c.s.AkkaHttpServer - Listening for HTTP on /0:0:0:0:0:0:0:0:9000
```

(Server started, use Enter to stop and go back to the console...)

```
[info] Compiling 6 Scala sources and 3 Java sources to /Users/mercedeswyss/Documents/Presentaciones/2017/OracleCode/Sydney/OracleCodeSydney/target/scala-2.12/classes...  
[info] p.a.h.EnabledFilters - Enabled Filters (see <https://www.playframework.com/documentation/latest/Filters>):
```

```
play.filters.csrf.CSRFFilter  
play.filters.headers.SecurityHeadersFilter  
play.filters.hosts.AllowedHostsFilter
```

```
[info] play.api.Play - Application started (Dev)
```



Project Structure

```
└─ .g8
└─ app
  └─ controllers
    └─ HomeController.java
  └─ views
    └─ index.scala.html
    └─ main.scala.html
└─ conf
└─ logs
└─ project
└─ public
└─ target
└─ test
└─ build.sbt
```

Directory	Description
app	The app and client side source directory controllers/ views/
conf	Config directory, route mapping, application config and log config
public	Static assets directory (images, css, js)
Test	Test source directory
project	sbt configuration files

main.scala.html

```
@*
 * This template is called from the `index` template. This template
 * handles the rendering of the page header and body tags. It takes
 * two arguments, a `String` for the title of the page and an `Html`
 * object to insert into the body of the page.
 *@
@(title: String)(content: Html)

<!DOCTYPE html>
<html lang="en">
  <head>
    @* Here's where we render the page title `String`. *@
    <title>@title</title>
    <link rel="stylesheet" media="screen" href="@routes.Assets.versioned("stylesheets/main.css")">
    <link rel="shortcut icon" type="image/png" href="@routes.Assets.versioned("images/favicon.png")">
  </head>
  <body>
    @* And here's where we render the `Html` object containing
    * the page content. *@
    @content

    <script src="@routes.Assets.versioned("javascripts/main.js")" type="text/javascript"></script>
  </body>
</html>
```



index.scala.html

```
@()
```

```
@main("Welcome to Play") {  
  <h1>Welcome to Play!</h1>  
}
```



HomeController.java

```
package controllers;

import play.mvc.*;

/**
 * This controller contains an action to handle HTTP requests
 * to the application's home page.
 */
public class HomeController extends Controller {

    /**
     * An action that renders an HTML page with a welcome message.
     * The configuration in the <code>routes</code> file means that
     * this method will be called when the application receives a
     * <code>GET</code> request with a path of <code>/</code>.
     */
    public Result index() {
        return ok(views.html.index.render());
    }
}
```



routes

Routes

This file defines all application routes (Higher priority routes first)

~~~~

An example controller showing a sample home page

GET / controllers.HomeController.index

Map static resources from the /public folder to the /assets URL path

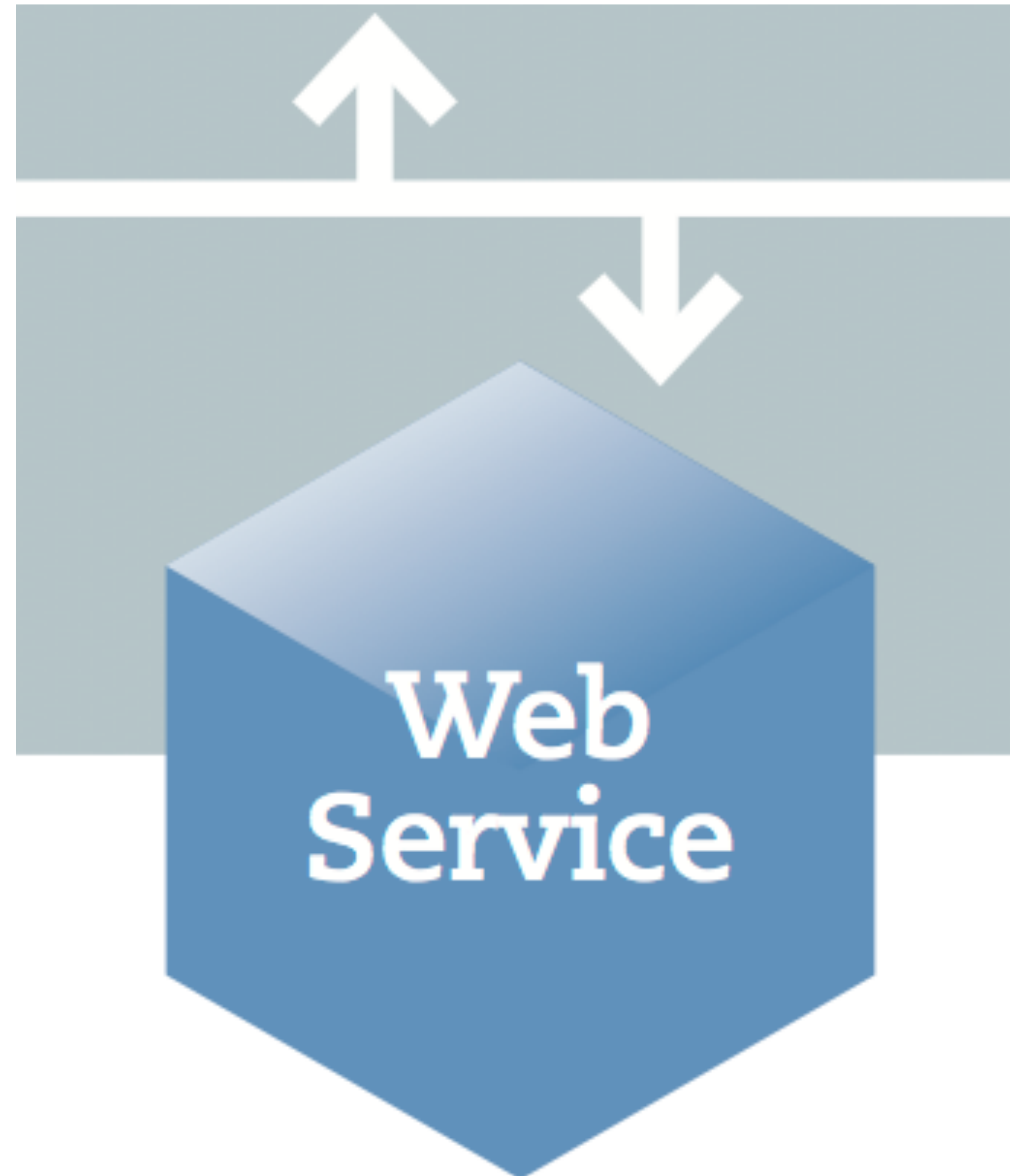
GET /assets/*file controllers.Assets.versioned(path="/public", file: Asset)



Agenda

- Play Framework
- **Web Services Restful**
- JSON
- JWT (JSON Web Tokens)

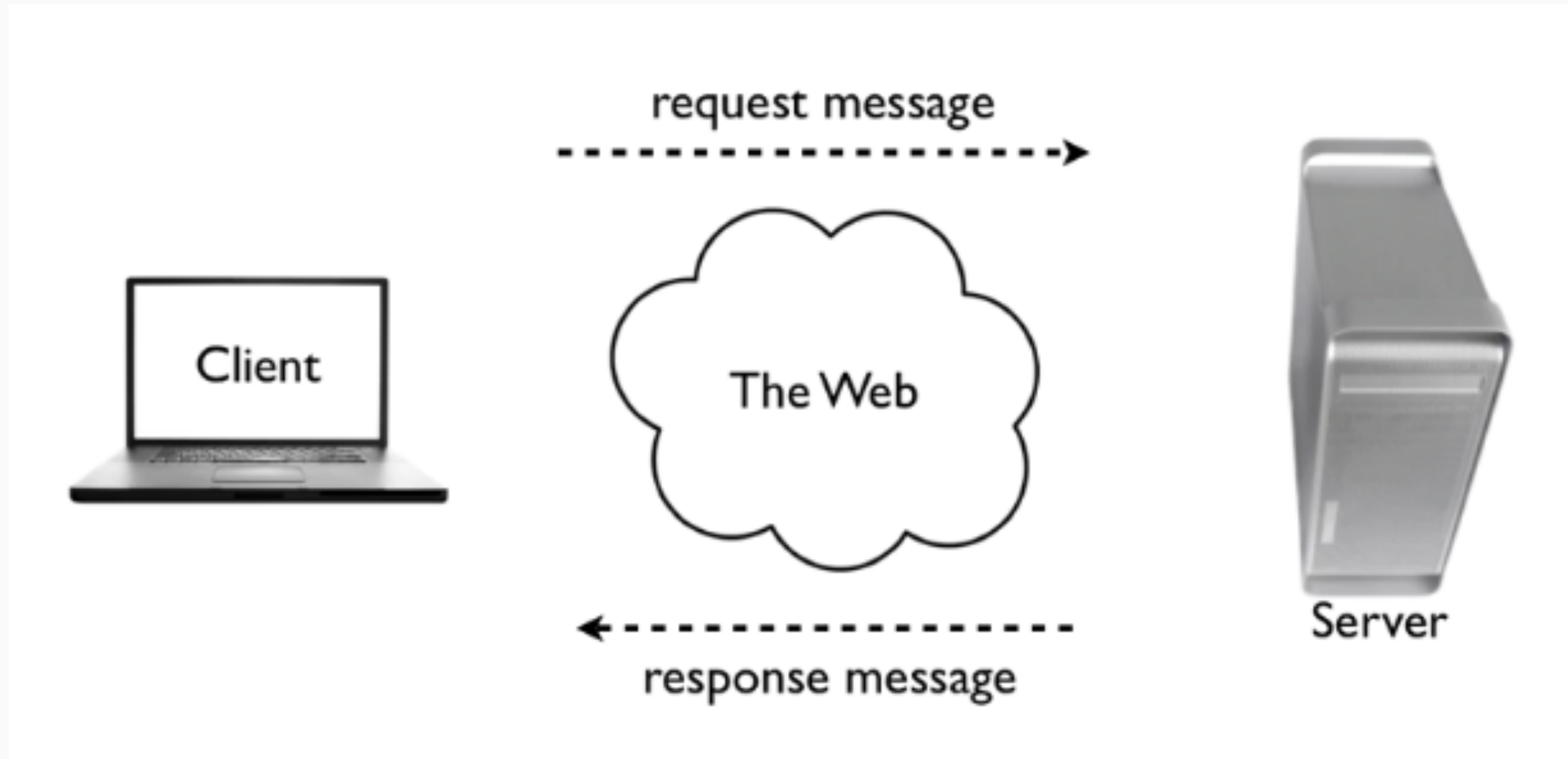




Web Service

- HyperText Transfer Protocol (HTTP)
- Allow communication (client \longleftrightarrow server)
- standard, interoperability, extensibility
- Based on XML



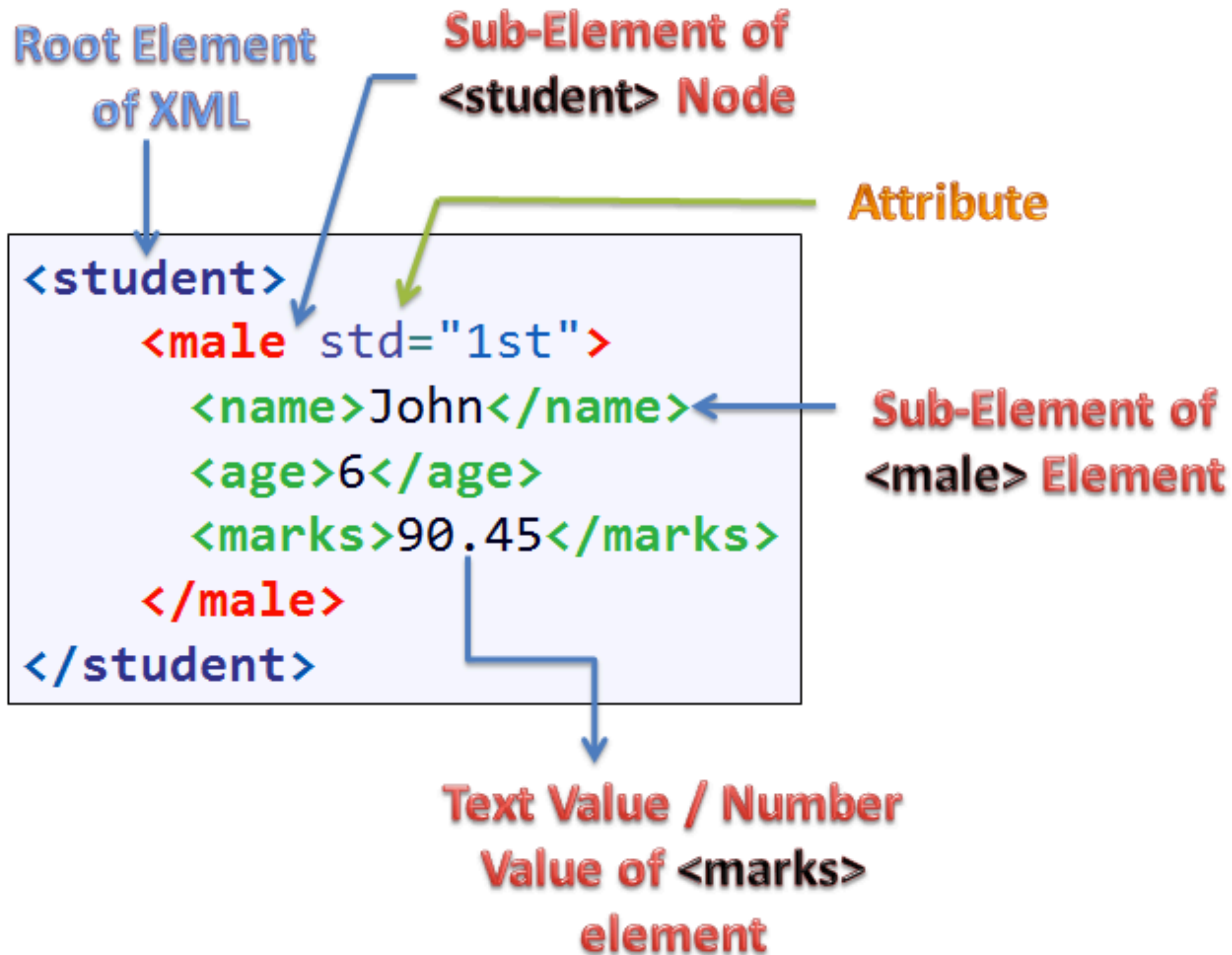


Linda.com (June 2017) <https://www.lynda.com/Software-Development-tutorials/What-web-service/126131/145941-4.html>

SOAP

- Simple Object Access Protocol
- XML-based
- WSDL as specification description
- Need to follow a communication protocol





Restful

- Representational State Transfer
- Software Architecture
- Based on HTTP methods
- Typically use JSON, can be XML or Text



SOAP vs REST

SOAP



REST



*REST are mostly used in industry

HomeController.java

```
package controllers;
```

```
import play.mvc.Controller;
```

```
import play.mvc.Result;
```

```
public class HomeController extends Controller {
```

```
    public Result index() {
```

```
        return ok(views.html.index.render());
```

```
    }
```

```
}
```



HomeController.java

```
package controllers;
```

```
import play.mvc.Controller;
```

```
import play.mvc.Result;
```

```
public class HomeController extends Controller {
```

```
    public Result index() {
```

```
        return ok(views.html.index.render());
```

```
    }
```

```
}
```



Simple.java

```
package controllers;

import play.mvc.Controller;
import play.mvc.Result;

public class Simple extends Controller {

    public Result text(){
        return ok("This is just a return message");
    }

}
```



Agenda

- Play Framework
- Web Services Restful
- **JSON**
- JWT (JSON Web Tokens)



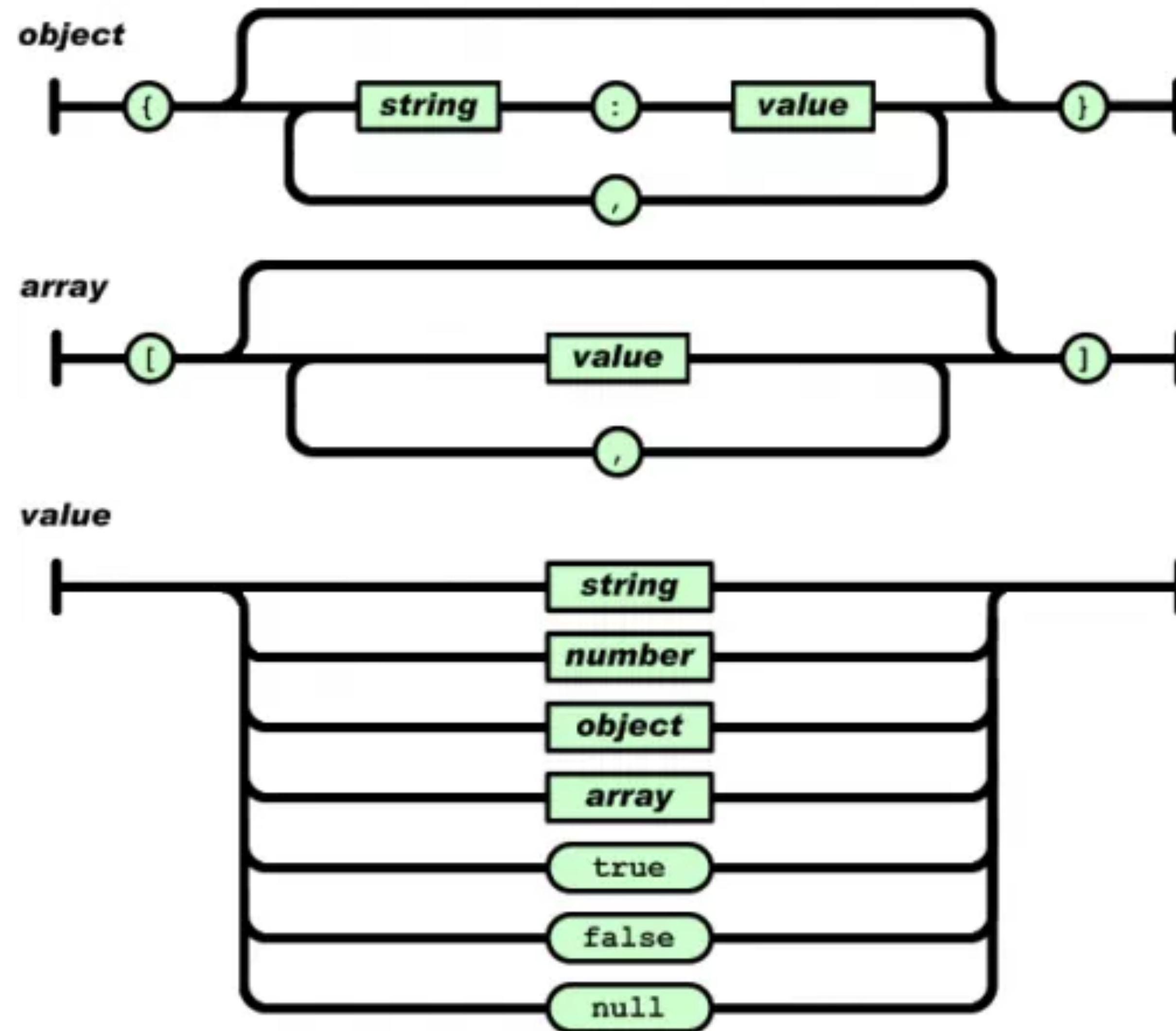
{JSON}

JSON

- JavaScript Object Notation
- Lightweight data-interchange format
- Self-describing
- Human-reading
- Is a Protocol



JSON Schema



JSON Example

```
{
  "Rail Booking": {
    "reservation": {
      "ref_no": 1234567,
      "time_stamp": "2016-06-24T14:26:59.125",
      "confirmed": true
    },
    "train": {
      "date": "07/04/2016",
      "time": "09:30",
      "from": "New York",
      "to": "Chicago",
      "seat": "57B"
    },
    "passenger": {
      "name": "John Smith"
    },
    "price": 1234.25,
    "comments": ["Lunch & dinner incl.", "\"Have a nice day!\""]
  }
}
```

JSONController.java

```
package controllers;

import play.libs.Json;
import play.mvc.Controller;
import play.mvc.Result;

import java.util.HashMap;
import java.util.Map;

public class JSONController extends Controller {

    public Result usingMap(){
        Map<String, Object> result = new HashMap<>();
        result.put("Event", "Oracle Developer Tour");
        result.put("Track", "Paraguay");
        result.put("Talk", "Restful Services with Play Framework");
        result.put("Schedule", "Miércoles 7 de Noviembre, 16:00hrs");

        return ok(Json.toJson(result));
    }
}
```



Event.java

```
package com.auth0.objects;
```

```
public class Event {
```

```
    public String event;
```

```
    public String track;
```

```
    public String talk;
```

```
    public String schedule;
```

```
    public Event(String event, String track, String talk, String schedule){
```

```
        this.event = event;
```

```
        this.track = track;
```

```
        this.talk = talk;
```

```
        this.schedule = schedule;
```

```
    }
```

```
}
```



JSONController.java

```
import com.auth0.objects.Event;
```

```
public Result usingObject(){  
    Event event = new Event("Oracle Developer Tour", "Paraguay",  
        "Restful Services with Play Framework",  
        "Miércoles 7 Noviembre, 16:00hrs");  
    return ok(Json.toJson(event));  
}
```



JSONController.java

```
public Result catchObject(){  
    JsonNode jsonNode = request().body().asJson();  
    Event event = Json.fromJson(jsonNode, Event.class);  
  
    return ok(Json.toJson(event));  
}
```



Agenda

- Play Framework
- Web Services Restful
- JSON
- **JWT (JSON Web Tokens)**



JWT

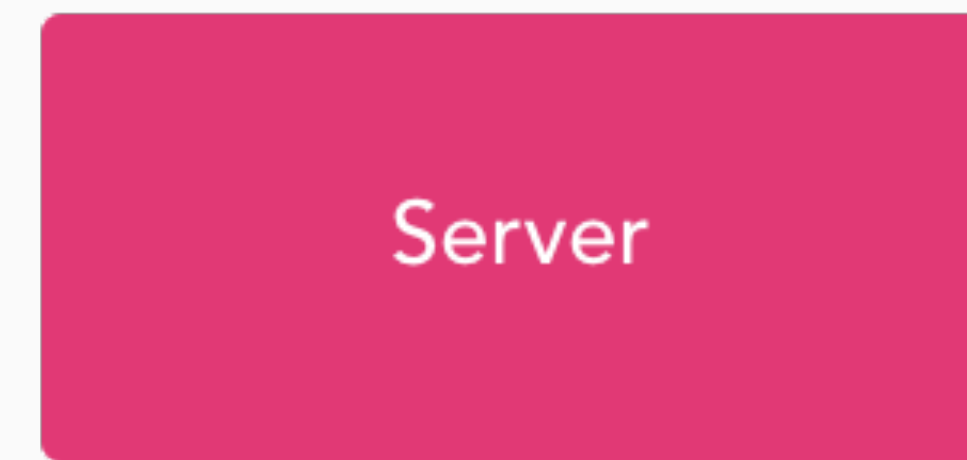
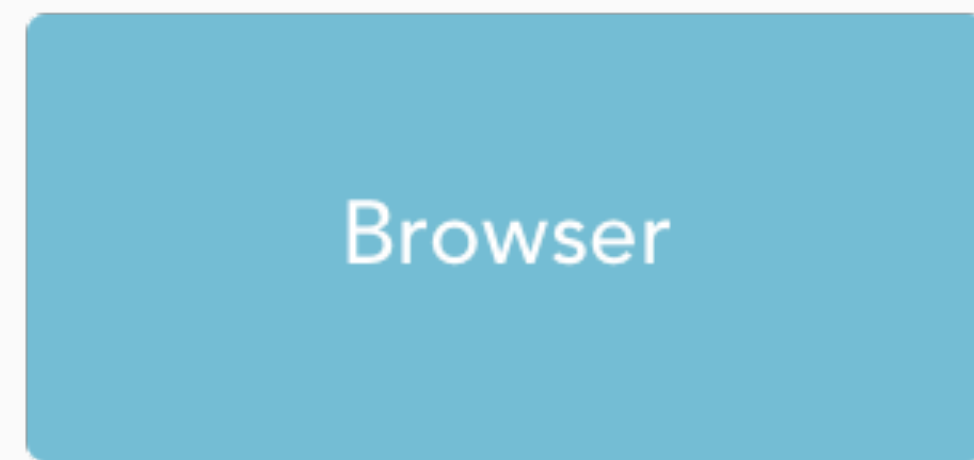
Is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object.



What problem does it solve?

- Authentication
- Authorization
- Federated Identity
- Information Exchange
- Client-side sessions ("stateless" sessions)
- Client-side secrets





1. POST /users/login with username and password

2. Creates a JWT with a secret

3. Returns the JWT to the Browser

4. Sends the JWT on the Authorization Header

5. Check JWT signature. Get user information from the JWT

6. Sends response to the client

Header

Claims

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleH
AiOjE0MTY0NzE5MzQsInVzZXJfbmFtZSI6InVzZXIiL
CJzY29wZSI6WyJyZWFKIiwid3JpdGUiXSwiYXV0aG9y
aXRpZXMia0lsiUk9MRV9BRE1JTjIIsImlJPTEVfVVNFUiJ
dLCJqdGkiOiI5YmM5MmE0NC0wYjFhLTRjNWUtYmU3MC
1kYTUyMDc1YjlhODQiLCJjbGllbnRfaWQiOiJteS1jb
GllbnQtd2l0aC1zZWNyZXQifQ.AZCTD_fiCcnrQR5X7
rJBQ5r0-2Qedc5_3qJJf-ZCvVY

Signature



ENCODED

PASTE A TOKEN HERE

DECODED

EDIT THE PAYLOAD AND SECRET (ONLY HS256 SUPPORTED)

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJhdWQiOiJ1Nm5uQXhHVmpiQmQ4ZXRYamo1N
TRZS0dBRzVIdVZycCI6InNjb3BlcyI6eyJ0b2
tlbnMiOnsiYWNoaW9ucyI6WyJibGFja2xpc3Q
iXX19LCJpYXQiOiJlMjUzZmZyZWYjImJcsImp0aSI6
IjA2MTI4ZDg5ZmRhOWEwYjAzN2JhZDkxZTJiM
WixN2RkIn0.YQzd4Y1weIRqDyR0FupXXmnp3k
3AzCmUALPKHNfn-xs

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

```
{
  "aud": "u6nnAxGVjbBd8etXjj554YKGAG5HuVrp",
  "scopes": {
    "tokens": {
      "actions": [
        "blacklist"
      ]
    }
  },
  "iat": 1425336027,
  "jti": "06128d89fda9a0b037bad91e2b1b17dd"
}
```



Auth0

Registered Claims

iss	The issuer of the token
sub	The subject of the token
aud	The audience of the token
exp	The expiration in NumericDate value
nbf	sbt configuration files
iat	The time the JWT was issued
jti	Unique identifier for the JWT

Registered Claims

iss	The issuer of the token
sub	The subject of the token
aud	The audience of the token
exp	The expiration in NumericDate value
nbf	sbt configuration files
iat	The time the JWT was issued
jti	Unique identifier for the JWT

Registered Claims

iss	The issuer of the token
sub	The subject of the token
aud	The audience of the token
exp	The expiration in NumericDate value
nbf	sbt configuration files
iat	The time the JWT was issued
jti	Unique identifier for the JWT

JSON Web Signature



Signature Algorithms

JWS	Algorithm	Description
HS256	HMAC256	HMAC with SHA-256
HS384	HMAC384	HMAC with SHA-384
HS512	HMAC512	HMAC with SHA-512
RS256	RSA256	RSASSA-PKCS1-v1_5 with SHA-256
RS384	RSA384	RSASSA-PKCS1-v1_5 with SHA-384
RS512	RSA512	RSASSA-PKCS1-v1_5 with SHA-512
ES256	ECDSA256	ECDSA with curve P-256 and SHA-256
ES384	ECDSA384	ECDSA with curve P-384 and SHA-384
ES512	ECDSA512	ECDSA with curve P-521 and SHA-512

```
name := """"JavaDayGT17""""  
organization := "com.auth0"
```

```
version := "1.0-SNAPSHOT"
```

```
lazy val root = (project in file(".")).enablePlugins(PlayJava)
```

```
scalaVersion := "2.12.2"
```

```
libraryDependencies += guice
```

```
libraryDependencies += "javax.xml.bind" % "jaxb-api" % "2.3.0"
```

```
libraryDependencies += "io.jsonwebtoken" % "jjwt" % "0.9.0"
```



Generando un JWT



JWTUtils.java

```
public static String generateJWT(final Object object){  
    String result = "";  
  
    try {  
        result = Jwts.builder()  
            .setClaims(new HashMap<String, Object>(){  
                put("json", Json.toJson(object));  
            })  
            .signWith(SignatureAlgorithm.HS256, "secre")  
            .compact();  
    } catch (ArrayIndexOutOfBoundsException e){  
        e.printStackTrace();  
    }  
  
    return result;  
}
```



JWTController.java

```
public Result returnToken(){
    Any any = new Any();
    any.level = Level.AVANZADOS;
    List<String> exito = new ArrayList<>();
    exito.add("valor 1");
    exito.add("valor 2");
    exito.add("valor 3");
    any.tags = exito;

    return ok(JWTUtils.generateJWT(any));
}
```



Validando un JWT



JWTUtils.java

```
public static boolean validateJWT(String jwt){
    boolean result = false;

    try {
        Claims claims = Jwts.parser()
            .setSigningKey(DatatypeConverter.parseBase64Binary("secre"))
            .parseClaimsJws(jwt).getBody();
        result = true;
    } catch (SignatureException e){
        e.printStackTrace();
    } catch (MalformedJwtException e){
        e.printStackTrace();
    } catch (ArrayIndexOutOfBoundsException e){
        e.printStackTrace();
    }

    return result;
}
```



JWTController.java

```
public Result validateToken(){  
    String token = request().body().asText();  
    return ok("Token Válido: " + JWTUtils.validateJWT(token));  
}
```





Debugger

ALGORITHM

HS256



Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjE5NjU0OTY0MDAwLCJ0eXAiOiJKWT0uTJVA950rM7E2cBab30RMHrHDcEfxjoYZgeF0NFh7HgQ
```

Decoded

EDIT THE PAYLOAD AND SECRET (ONLY HS256 SUPPORTED)

HEADER: ALGORITHM & TOKEN TYPE

```
{  "alg": "HS256",  "typ": "JWT"}
```

PAYLOAD: DATA

```
{  "sub": "1234567890",
```



Learn everything you wanted
to know, but were afraid to
ask about JSON Web Tokens

auth0.com/e-books/jwt-handbook

<https://github.com/itrjwyss/ODT2017>

<https://www.facebook.com/itrjwyss>

@itrjwyss

