

Javalin

Simple REST APIs for Java and Kotlin

Get your REST API up and running in seconds. Add the [dependency](#) and copy example:

```
Java Kotlin Copy
import io.javalin.Javalin;

public class HelloWorld {
    public static void main(String[] args) {
        Javalin app = Javalin.start(7000);
        app.get("/", ctx -> ctx.result("Hello World"));
    }
}
```

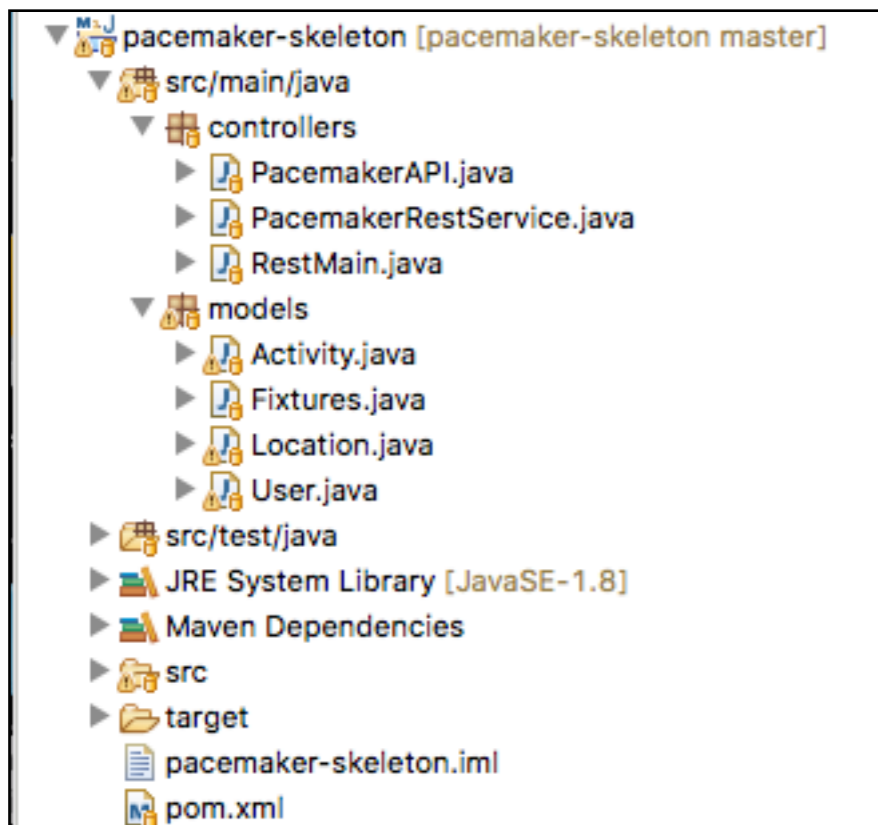
REST API Simplicity

Javalin started as a fork of the [Spark](#) Java and Kotlin web framework but quickly turned into a ground-up rewrite influenced by [koa.js](#). All of these web frameworks are inspired by the modern micro web framework grandfather: [Sinatra](#), so if you're coming from Ruby then Javalin shouldn't feel *too* unfamiliar.

Javalin is not aiming to be a full web framework, but rather just a lightweight REST API library. There is no concept of MVC, but there is support for template engines, websockets, and static file serving for convenience. This allows you to use Javalin for both creating your RESTful API backend, as well as serving an `index.html` with static resources (in case you're creating an SPA). This is practical if you don't want to deploy to apache or nginx in addition to your Javalin service. If you wish to use Javalin to create web-pages instead of just REST APIs, there are some simple template engine wrappers available for a quick and easy setup.

pacemaker-skeleton

Extend project to include Javalin components



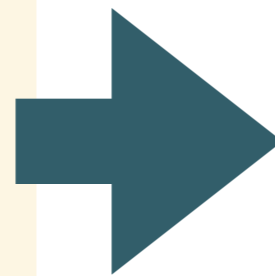
```
<dependency>  
  <groupId>io.javalin</groupId>  
  <artifactId>javalin</artifactId>  
  <version>0.5.1</version>  
</dependency>  
<dependency>  
  <groupId>org.slf4j</groupId>  
  <artifactId>slf4j-simple</artifactId>  
  <version>1.7.21</version>  
</dependency>  
<dependency>  
  <groupId>com.fasterxml.jackson.core</groupId>  
  <artifactId>jackson-databind</artifactId>  
  <version>2.9.1</version>  
</dependency>
```

pom.xml

Eclipse downloads and injects range of components to support creation of REST services, including

- web server
- http protocol support
- json/java translation

```
<dependency>
  <groupId>io.javalin</groupId>
  <artifactId>javalin</artifactId>
  <version>0.5.1</version>
</dependency>
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-simple</artifactId>
  <version>1.7.21</version>
</dependency>
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-databind</artifactId>
  <version>2.9.1</version>
</dependency>
```

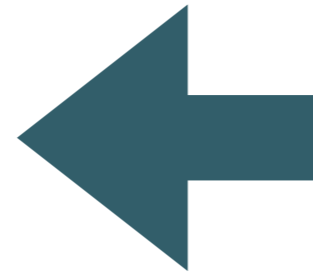
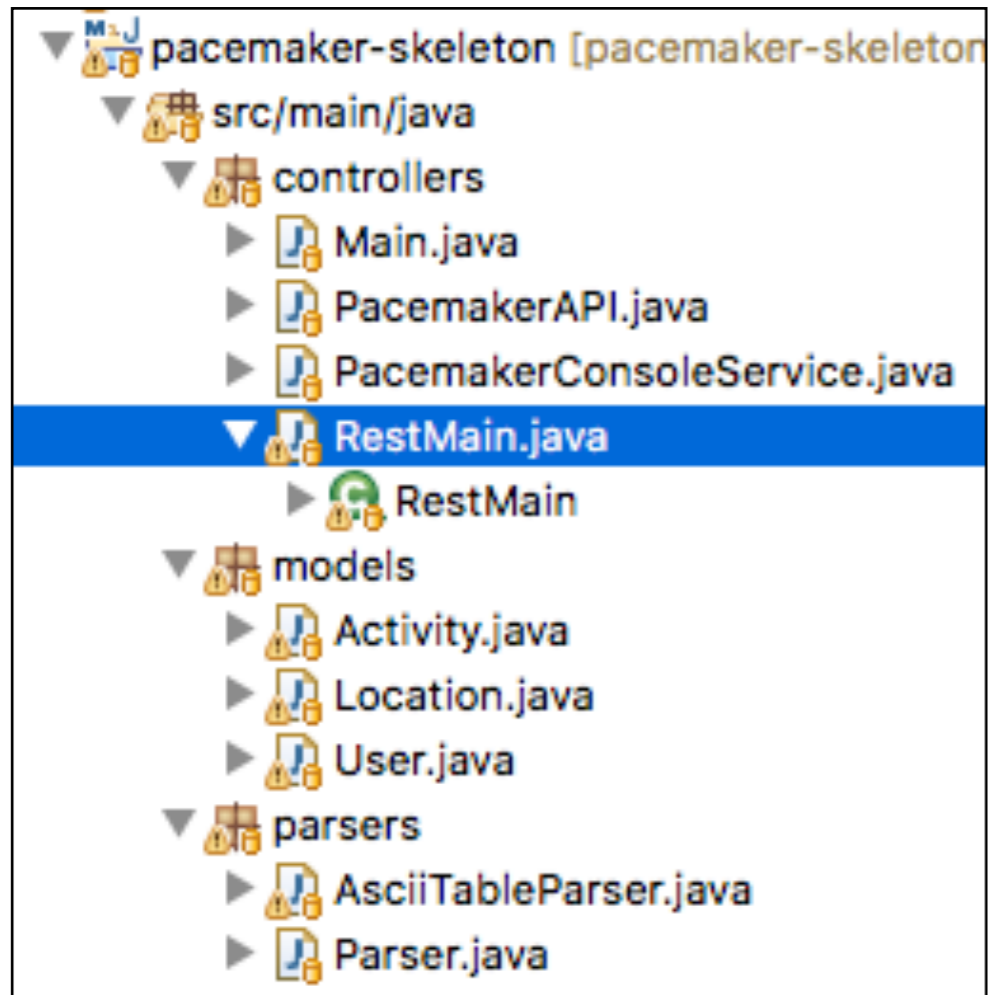


The screenshot shows the Eclipse IDE interface for a project named 'pacemaker-skeleton'. The project structure is as follows:

- src/main/java
 - controllers
 - Main.java
 - PacemakerAPI.java
 - PacemakerConsoleService.java
 - PacemakerRestService.java
 - models
 - Activity.java
 - Location.java
 - User.java
 - parsers
 - AsciiTableParser.java
 - Parser.java
- src/test/java

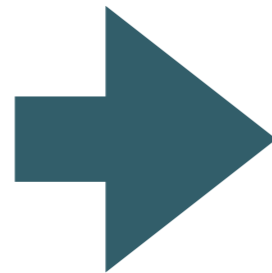
Below the source code, the 'Maven Dependencies' section is expanded, showing a list of downloaded JAR files:

- junit-4.12.jar
- hamcrest-core-1.3.jar
- guava-23.0.jar
- jsr305-1.3.9.jar
- error_prone_annotations-2.0.18.jar
- j2objc-annotations-1.1.jar
- animal-sniffer-annotations-1.14.jar
- asg-cliche-1.0.jar
- java-ascii-table-1.0.jar
- javalin-0.5.1.jar
- kotlin-stdlib-jre8-1.1.2-2.jar
- kotlin-stdlib-jre7-1.1.2-2.jar
- kotlin-stdlib-1.1.2-2.jar
- annotations-13.0.jar
- slf4j-api-1.7.25.jar
- jetty-server-9.4.6.v20170531.jar
- javax.servlet-api-3.1.0.jar
- jetty-http-9.4.6.v20170531.jar
- jetty-util-9.4.6.v20170531.jar
- jetty-io-9.4.6.v20170531.jar
- jetty-webapp-9.4.6.v20170531.jar
- jetty-xml-9.4.6.v20170531.jar
- jetty-servlet-9.4.6.v20170531.jar
- jetty-security-9.4.6.v20170531.jar
- jackson-databind-2.9.1.jar
- jackson-annotations-2.9.0.jar
- jackson-core-2.9.1.jar



Leave existing
application sources
in place

Introduce a
new main()
method to start
the application



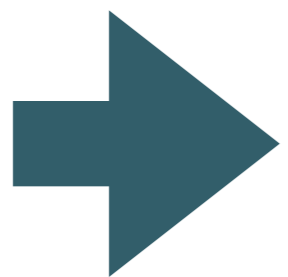
```
package controllers;
import io.javalin.Javalin;
public class RestMain {
    public static void main(String[] args) throws Exception {
        Javalin app = Javalin.start(7000);
    }
}
```



```
Console [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_60.jdk/Contents/Home/bin/java (20 Nov 2017, 09:47:34)

[main] INFO io.javalin.Javalin - Starting Javalin
[main] INFO org.eclipse.jetty.server.Server - jetty
[main] INFO org.eclipse.jetty.server.session - Def
[main] INFO org.eclipse.jetty.server.session - No
[main] INFO org.eclipse.jetty.server.session - Sco
[main] INFO org.eclipse.jetty.server.handler.Conte
[main] INFO org.eclipse.jetty.server.AbstractConne
[main] INFO org.eclipse.jetty.server.Server - Star
[main] INFO io.javalin.embeddedserver.EmbeddedServ
[main] INFO io.javalin.Javalin - Javalin has start
```

Web Application
(which will serve
an API) running
on local
workstation



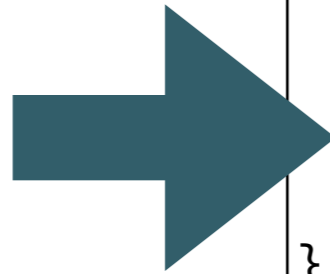
localhost:7000

Not found

Recording network activity...
Perform a request or hit ⌘ R to record the reload.

Failed to load resource: the server responded with a status of 404 (Not Found)

Extend main to
define REST API



```
package controllers;

import io.javalin.Javalin;

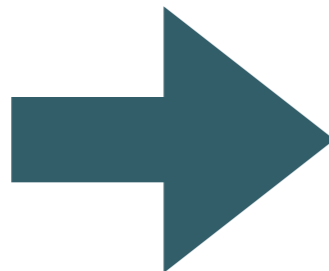
public class RestMain {

    public static void main(String[] args) throws Exception {
        Javalin app = Javalin.start(7000);
        PacemakerRestService service = new PacemakerRestService();
        configRoutes(app, service);
    }

    static void configRoutes(Javalin app, PacemakerRestService service) {

        app.get("/users", ctx -> {
            service.listUsers(ctx);
        });
    }
}
```

First version of
API (stub)



```
package controllers;

import io.javalin.Context;

public class PacemakerRestService {

    PacemakerAPI pacemaker = new PacemakerAPI();

    public void listUsers(Context ctx) {
        // TODO: return users in json format
    }
}
```



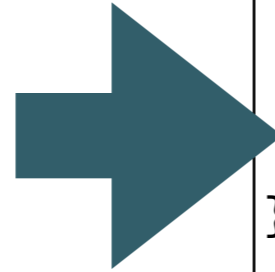
```
package controllers;

import io.javalin.Context;

public class PacemakerRestService {

    PacemakerAPI pacemaker = new PacemakerAPI();

    public void listUsers(Context ctx) {
        // TODO: return users in json format
    }
}
```



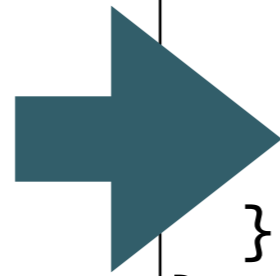
The screenshot shows a web browser window with the address bar displaying 'localhost:7000/users'. The browser's developer tools are open, and the Network tab is selected. A single request is visible in the network log:

Name	Status	Type	Initiator	Size	Time	Waterfall
users	200	document	Other	132 B	8 ms	

Summary: 1 requests | 132 B transferred | Finish: 8 ms | DOMContentLoaded: 16 ms | Load: 16 ms

listUsers ()
should query
pacemakerAPI
and return
users in json
format

Fetch list of users
from pacemaker.
Convert list to
JSON.
Send JSON list as
response to client



```
package controllers;

import io.javalin.Context;

public class PacemakerRestService {

    PacemakerAPI pacemaker = new PacemakerAPI();

    public void listUsers(Context ctx) {

        ctx.json(pacemaker.getUsers());

    }

}
```

```
ctx.json(object); // set result with object-as-json (requires jackson)
```

localhost:7000/users

localhost:7000/users

```
[{"id": "a365b857-f5c7-45dc-9255-81ba7d2d6523", "firstName": "marge", "lastName": "simpson", "email": "marge@simpson.com", "password": "secret", "activities": {}}, {"id": "04416a7f-c8f4-4462-87d6-ab5da2529a25", "firstName": "lisa", "lastName": "simpson", "email": "lisa@simpson.com", "password": "secret", "activities": {}}, {"id": "4389d2fb-18c7-43b4-a6ed-3d52d55a315a", "firstName": "bart", "lastName": "simpson", "email": "bart@simpson.com", "password": "secret", "activities": {}}, {"id": "f0662dae-21fe-40d6-a47a-3fc45847f738", "firstName": "maggie", "lastName": "simpson", "email": "maggie@simpson.com", "password": "secret", "activities": {}}]
```

Elements Console Sources Network Performance Memory Application Security Audits

View: Group by frame Preserve log Disable cache Offline Online

Filter Regex Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other

Name	Status	Type	Initiator	Size	Time	Waterfall	100.00
users	200	document	Other	892 B	418 ms		

1 requests | 892 B transferred | Finish: 418 ms | DOMContentLoaded: 433 ms | Load: 427 ms

Console What's New

top Filter Default levels

Restlet Client - REST API Testing | chrome-extension://aejoelaoggembcahagimdiliamlcdmfm/restlet_client.html

CLIENT REQUESTS SCENARIOS Settings Pricing Help Sign in

HISTORY REPOSITORY

Search

MY DRIVE
NO SAVED DATA
SIGN IN TO ACCESS SHARED PROJECTS

type a name Save 2 Code Reset Add an environment

REQUEST

METHOD: GET | SCHEME://HOST [":" PORT][PATH ["?" QUERY]]

http://localhost:7000/users length: 27 bytes

Send

QUERY PARAMETERS

HEADERS Form BODY

+ Add header Add authorization

XHR does not allow payloads for GET request. or change a method definition in settings.

RESPONSE Cache Detected - Elapsed Time: 5ms

200 OK

HEADERS pretty BODY pretty

Date: 2017 Oct 14 17:13:17
Server: Javalin
Content-Type: application/json
Content-Length: 603 bytes

COMPLETE REQUEST HEADERS pretty

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36

Accept: */*
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.8,nb;q=0.6,ga;q=0.4,la;q=0.2

```
[
  {
    id: "c4d90362-d9a1-4e41-84e0-7852bee865fd",
    firstname: "bart",
    lastname: "simpson",
    email: "bart@simpson.com",
    password: "secret",
    activities: {}
  },
  {
    id: "071cbd70-3ef5-48b5-b2ed-f1614ef7cc9a",
    firstname: "maggie",
    lastname: "simpson",
    email: "maggie@simpson.com",
    password: "secret",
    activities: {}
  }
]
```

Top Bottom Collapse Open 2Request Copy Download

HISTORY ASSERTIONS HTTP DESCRIPTION TIPS

GET /users HTTP/1.1
Host: localhost:7000

HTTP/1.1 200 OK
Date: Sat, 14 Oct 2017 16:13:17 GMT
Server: Javalin
Content-Type: application/json
Content-Length: 603

```
{["id":"c4d90362-d9a1-4e41-84e0-7852bee865fd","firstname":"bart","lastname":"simpson","email":"bart@simpson.com","password":"secret",
```

Export Import Add

pacemaker Endpoints

GET	/users	: read all users
POST	/users	: create a new user
GET	/users/id	: get a specific user (by id)
GET	/users/id/activities	: get activities for specific user (by id)
POST	/users/id/activities	: create activity for specific user (by id)

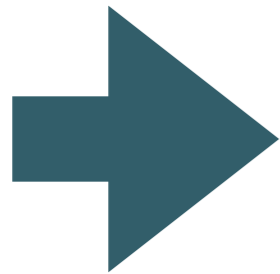
Endpoints deliver API for user
and activity resources

Access to API from any REST
away client

GET	/users	: read all users
POST	/users	: create a new user
GET	/users/id	: get a specific user (by id)
GET	/users/id/activities	: get activities for specific user (by id)
POST	/users/id/activities	: create activity for specific user (by id)

Endpoint Interface

Routes:
here we define
the supported
endpoints



```
static void configRoutes(Javalin app, PacemakerRestService service) {  
  
    app.get("/users", ctx -> {  
        service.listUsers(ctx);  
    });  
  
    app.post("/users", ctx -> {  
        service.createUser(ctx);  
    });  
  
    app.get("/users/:id", ctx -> {  
        service.listUser(ctx);  
    });  
  
    app.get("/users/:id/activities", ctx -> {  
        service.getActivities(ctx);  
    });  
  
    app.post("/users/:id/activities", ctx -> {  
        service.createActivity(ctx);  
    });  
  
}
```

Create User Endpoint

POST /users : create a new user

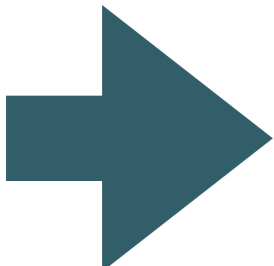
Endpoint Interface

```
app.post("/users", ctx -> {  
    service.createUser(ctx);  
});
```

Endpoint Implementation

```
public void createUser(Context ctx) {  
    User user = ctx.bodyAsClass(User.class);  
    User newUser = pacemaker  
        .createUser(user.firstname, user.lastname, user.email, user.password);  
    ctx.json(newUser);  
}
```

Formulate
POST
request



REQUEST

METHOD: POST SCHEME :// HOST [":" PORT] [PATH ["?" QUERY]]
http://localhost:7000/users length: 27 bytes **Send**

QUERY PARAMETERS

HEADERS [?] ↓½ Form ▾ BODY [?] Text ▾

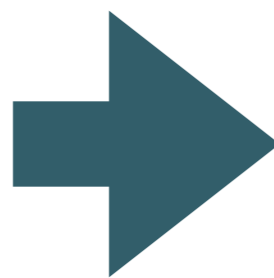
Cont : application/json ×

+ Add header Add authorization

```
1 {
2   "firstname" : "homer",
3   "lastname" : "Simpson",
4   "email" : "homer@simpson.com",
5   "password" : "secret"
6 }
```

Testing
the
Endpoint

Inspect
response



200 OK

HEADERS [?] pretty ▾ BODY [?] pretty ▾

Date: 2017 Oct 14 17:22:59
Server: Javalin
Content-T... application/json
Content-L... 150 bytes

COMPLETE REQUEST HEA... pretty ▾

Origin: chrome-extension://aejoelaoggembcahagimdiliamlcdmfm
User-Agen... Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36

Content-T... application/json
Accept: */*
Accept-En... gzip, deflate, br
Accept-La... en-US,en;q=0.8,nb;q=0.6,ga;q=0.4,la;q=0.2

Note: XHR automatically adds headers like Accept, Accept-Language, Cookie, User-Agent, etc.

```
{
  id: "105f8db6-ec5c-4740-b504-0cb64e6c468a",
  firstname: "homer",
  lastname: "Simpson",
  email: "homer@simpson.com",
  password: "secret",
  activities: {}
}
```

length: 150 bytes


```
public void createUser(Context ctx) {  
    User user = ctx.bodyAsClass(User.class);  
    User newUser = pacemaker  
        .createUser(user.firstname, user.lastname, user.email, user.password);  
    ctx.json(newUser);  
}
```

```
▼ {  
  id: "105f8db6-ec5c-4740-b504-0cb64e6c468a",  
  firstname: "homer",  
  lastname: "Simpson",  
  email: "homer@simpson.com",  
  password: "secret",  
  activities: ▶ {}  
}
```

Covert incoming JSON object to Java Object

```
public void createUser(Context ctx) {  
    User user = ctx.bodyAsClass(User.class);  
  
    User newUser = pacemaker  
        .createUser(user.firstname, user.lastname, user.email, user.password);  
  
    ctx.json(newUser);  
}
```

Create object using pacemaker

Convert new user object (including id) to json and return to client

```
{  
  id: "105f8db6-ec5c-4740-b504-0cb64e6c468a",  
  firstname: "homer",  
  lastname: "Simpson",  
  email: "homer@simpson.com",  
  password: "secret",  
  activities: {}  
}
```

POST /users/id/activities : create activity for specific user (by id)

```
app.post("/users/:id/activities", ctx -> {  
  service.createActivity(ctx);  
});
```

```
public void createActivity(Context ctx) {  
  String id = ctx.param("id");  
  User user = pacemaker.getUser(id);  
  if (user != null) {  
    Activity activity = ctx.bodyAsClass(Activity.class);  
    Activity newActivity = pacemaker  
      .createActivity(id, activity.type, activity.location, activity.distance);  
    ctx.json(newActivity);  
  } else {  
    ctx.status(404);  
  }  
}
```

POST /users/:id/activities : create activity for specific user (by id)

```
app.post("/users/:id/activities", ctx -> {  
  service.createActivity(ctx);  
});
```

Retrieve user id from url

```
public void createActivity(Context ctx) {  
  String id = ctx.param("id");  
  User user = pacemaker.getUser(id);  
  if (user != null) {  
    Activity activity = ctx.bodyAsClass(Activity.class);  
    Activity newActivity = pacemaker  
      .createActivity(id, activity.type, activity.location, activity.distance);  
    ctx.json(newActivity);  
  } else {  
    ctx.status(404);  
  }  
}
```

Lookup user in
pacemaker

Recover activity
payload from request

Return 'not found' if no
mathing user (by id)

Create new activity &
return it to client