#### Agile Software Development



Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics Waterford Institute of Technology

http://www.wit.ie

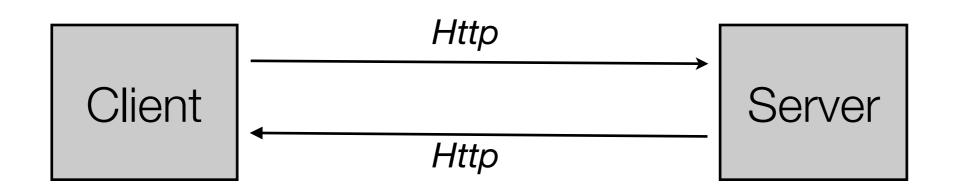
http://elearning.wit.ie





# Play 2

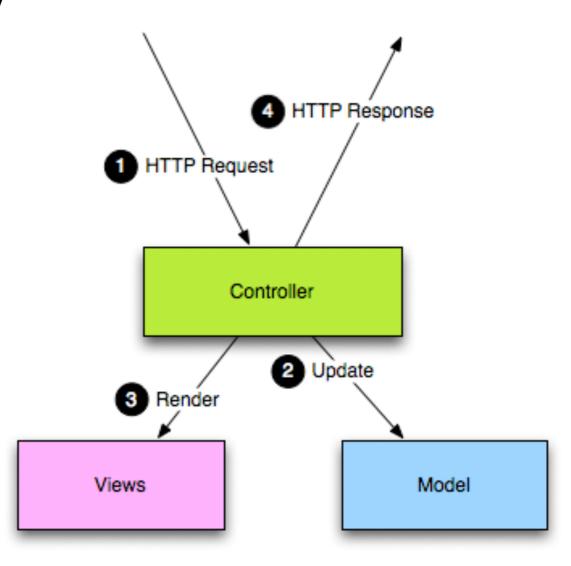
#### Web Applications - Request/Response



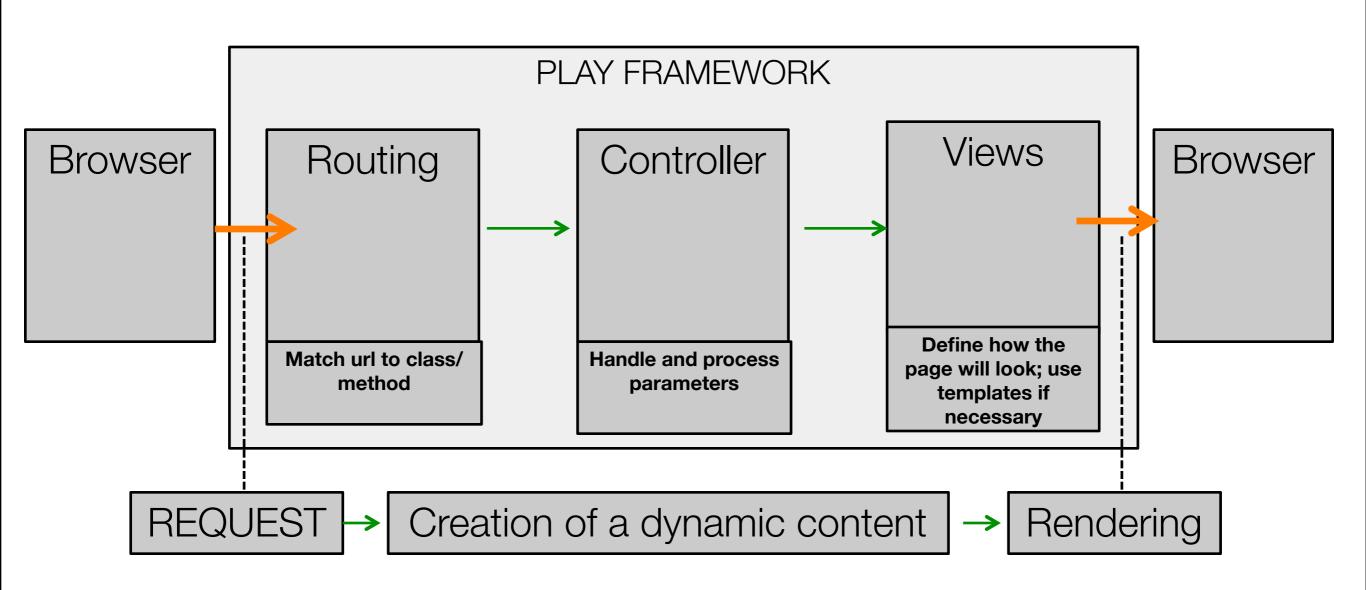
- Request http request emitted by browser as a result of url in address bar, link, button or form submission on page
- Response web page returned from service to be presented in browser

## Web Applications - MVC

- Model View Controller is a generally accepted pattern or separation of concerns within the server
- Model: Core application domain model + database persistence
- View: User Experience
- Controller: Directly handle all requests, mediate with Model, build and assemble the response using the views

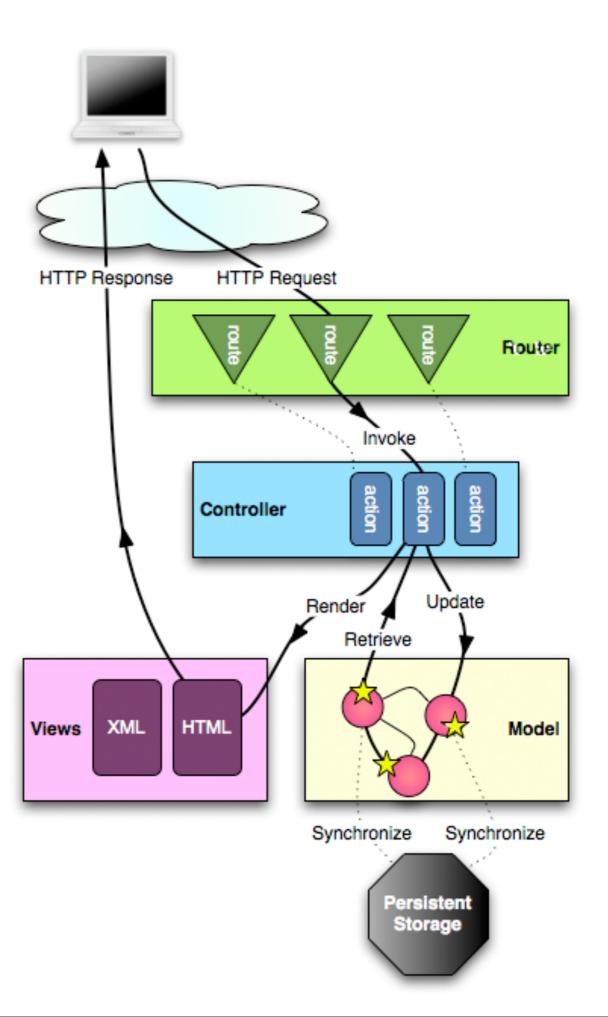


# Request/Response Lifecycle



## MVC in Play

- Router: examine incoming requests and match to corresponding Controller/Action
- Action: a method in the controller



## Routes File Example (Pacemaker)

```
# UI
GET
                                                    controllers.Accounts.index()
        /signup
                                                    controllers.Accounts.signup()
GET
GET
        /login
                                                   controllers.Accounts.login()
                                                    controllers.Accounts.logout()
GET
        /logout
        /register
                                                   controllers.Accounts.register()
POST
        /authenticate
                                                   controllers.Accounts.authenticate()
POST
                                                   controllers.Dashboard.index()
GET
        /dashboard
        /upload
                                                    controllers.Dashboard.uploadActivityForm()
GET
        /submitactivity
                                                    controllers.Dashboard.submitActivity()
POST
# API
GET
        /api/users
                                                   controllers.PacemakerAPI.users()
DELETE /api/users
                                                    controllers.PacemakerAPI.deleteAllUsers()
        /api/users
                                                   controllers.PacemakerAPI.createUser()
POST
GET
       /api/users/:id
                                                   controllers.PacemakerAPI.user(id: Long)
DELETE /api/users/:id
                                                   controllers.PacemakerAPI.deleteUser(id: Long)
      /api/users/:id
                                                   controllers.PacemakerAPI.updateUser(id: Long)
PUT
        /api/users/:userId/activities
                                                   controllers.PacemakerAPI.activities(userId: Long)
GET
        /api/users/:userId/activities
                                                   controllers.PacemakerAPI.createActivity(userId: Long)
POST
        /api/users/:userId/activities/:activityId
                                                   controllers. Pacemaker API. activity (userId: Long, activityId:Long)
GET
DELETE /api/users/:userId/activities/:activityId
                                                   controllers.PacemakerAPI.deleteActivity(userId: Long, activityId:Long)
        /api/users/:userId/activities/:activityId controllers.PacemakerAPI.updateActivity(userId: Long, activityId:Long)
PUT
# Map static resources from the /public folder to the /assets URL path
        /assets/*file
                                    controllers.Assets.at(path="/public", file)
GET
```

#### Routes - UI

```
# UI
                                                    controllers.Accounts.index()
GET
GET
       /signup
                                                    controllers.Accounts.signup()
GET
        /login
                                                    controllers.Accounts.login()
GET
       /logout
                                                    controllers.Accounts.logout()
       /register
                                                    controllers.Accounts.register()
POST
POST
        /authenticate
                                                    controllers.Accounts.authenticate()
        /dashboard
                                                    controllers.Dashboard.index()
GET
       /upload
                                                    controllers.Dashboard.uploadActivityForm()
GET
        /submitactivity
                                                    controllers.Dashboard.submitActivity()
POST
```

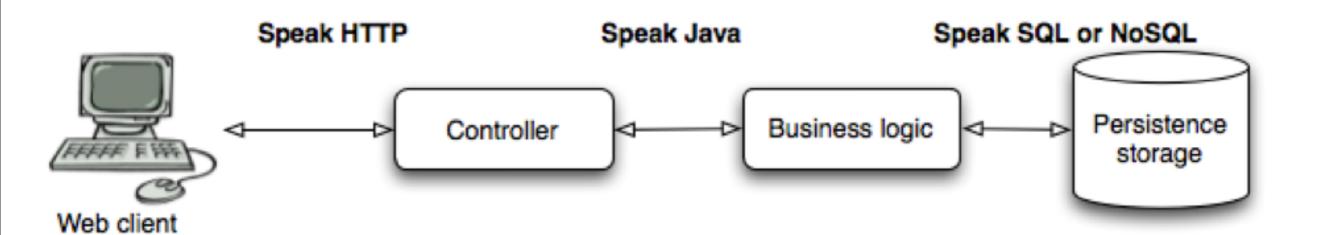
- Routes to deliver UI
- Each of these routes appears in views
- Each of these actions generates and returns a complete HTML page

#### Routes - API

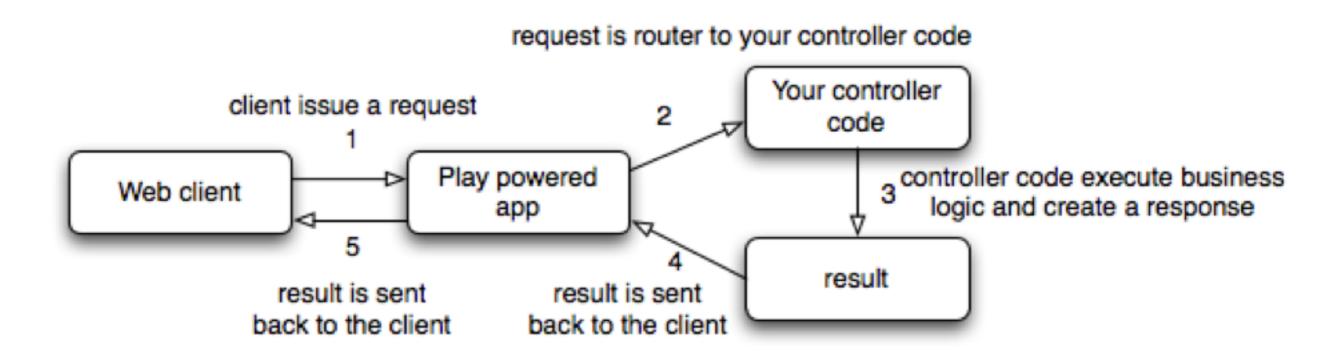
```
# API
                                                     controllers.PacemakerAPI.users()
        /api/users
GET
DELETE /api/users
                                                     controllers.PacemakerAPI.deleteAllUsers()
        /api/users
                                                     controllers.PacemakerAPI.createUser()
P<sub>0</sub>ST
       /api/users/:id
                                                     controllers.PacemakerAPI.user(id: Long)
GET
DELETE /api/users/:id
                                                     controllers.PacemakerAPI.deleteUser(id: Long)
PUT
       /api/users/:id
                                                     controllers.PacemakerAPI.updateUser(id: Long)
        /api/users/:userId/activities
                                                     controllers.PacemakerAPI.activities(userId: Long)
GET
        /api/users/:userId/activities
                                                     controllers.PacemakerAPI.createActivity(userId: Long)
POST
        /api/users/:userId/activities/:activityId
                                                    controllers. Pacemaker API. activity (userId: Long, activityId:Long)
GET
DELETE /api/users/:userId/activities/:activityId
                                                    controllers. Pacemaker API. delete Activity (user Id: Long, activity Id: Long)
        /api/users/:userId/activities/:activityId
                                                    controllers.PacemakerAPI.updateActivity(userId: Long, activityId:Long)
PUT
```

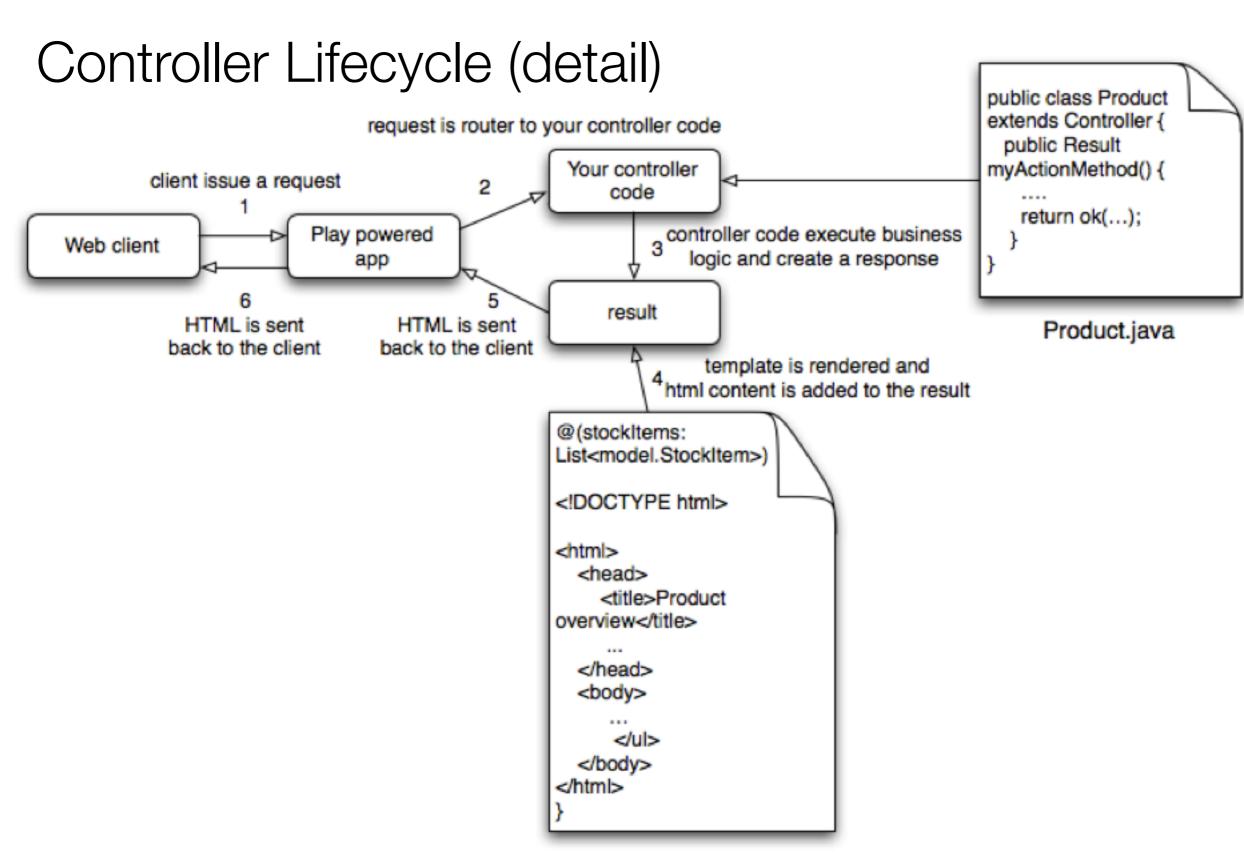
- Routes to deliver API
- Each of these routes appears in views an API Request
- Each of these actions generates and returns a JSON payload

#### Role of Controller



# Controller Lifecycle



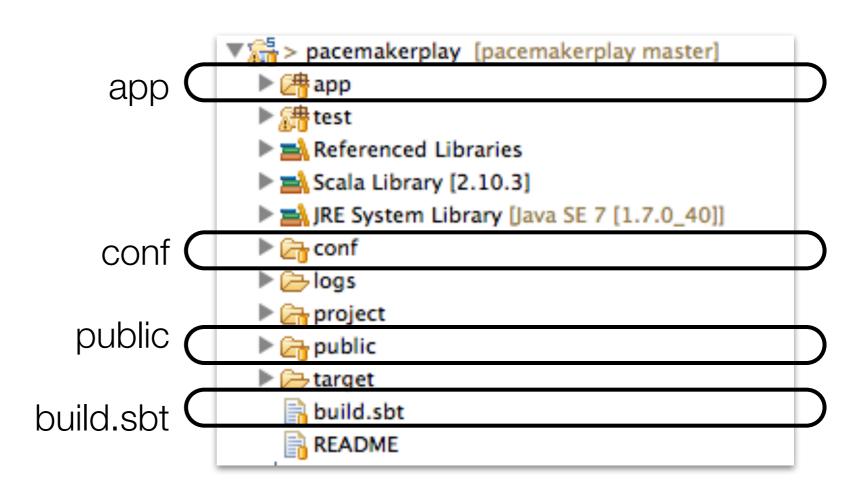


product.scala.html

# Controller Lifecycle with Content

#### request is router to your controller code Your controller client issue a request 2 code 3 controller code execute business Play powered Web client logic and create a result app result result is sent result is sent back to the client back to the client content is added to the result (text, html or whatever you like) content

# Play Project Anatomy



# App

▼ 🌁 app ▼ 

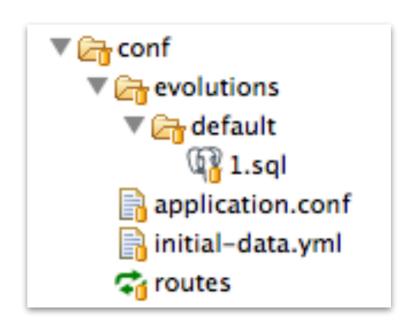
⊕ (default package) ▶ ☐ Global.java ▼ 

⊕ controllers controllers Accounts.java Dashboard.java PacemakerAPI.java ▼ 

models models Activity.java User.java parsers ▶ In JsonParser.java views **▼** thews accounts\_login.scala.html accounts\_signup.scala.html dashboard\_main.scala.html dashboard\_uploadactivity.scala.html main.scala.html welcome\_main.scala.html

welcome\_menu.scala.html

#### conf



# public

#### build.sbt

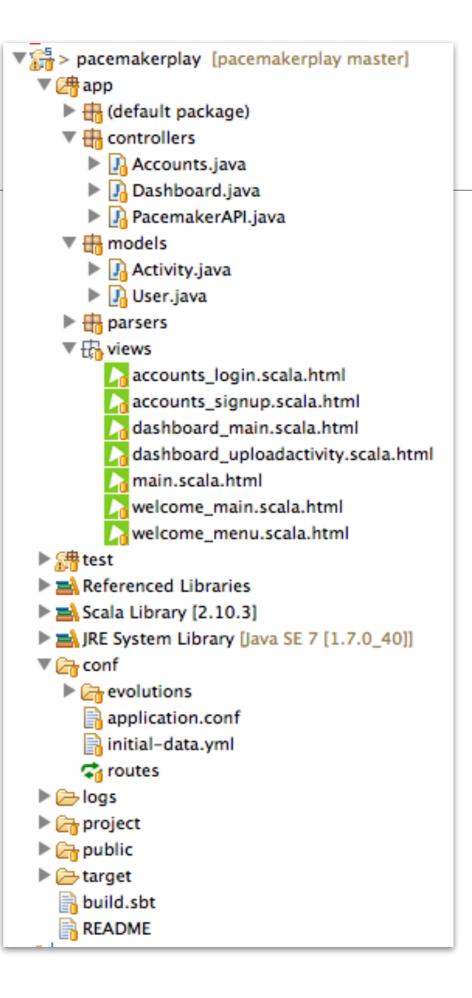
```
name := "pacemakerplay"

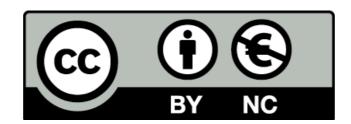
version := "1.0-SNAPSHOT"

libraryDependencies ++= Seq(
    javaJdbc,
    javaEbean,
    cache,
    "net.sf.flexjson" % "flexjson" % "3.1",
    "postgresql" % "postgresql" % "9.1-901-1.jdbc4"
)

play.Project.playJavaSettings
```

# pacemakerplay





Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see http://creativecommons.org/licenses/by-nc/3.0/



