



AFS

Agile Full Stack
Developer Bootcamp
Thoughtworks @

Pair Programming

/thoughtworks



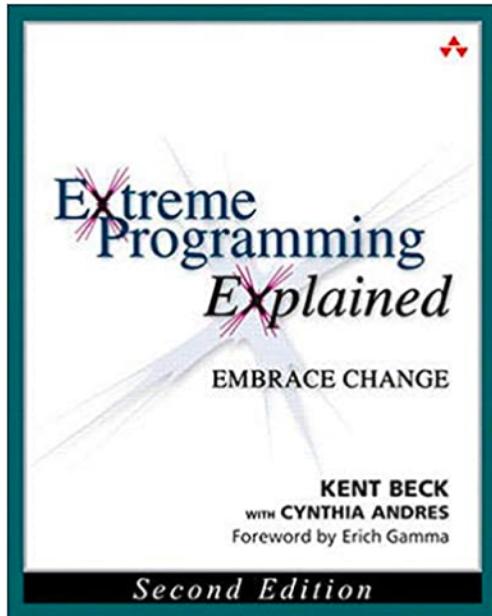
AFS

Agile Full Stack
Developer Bootcamp
Thoughtworks @

What is Pair Programming?

/thoughtworks

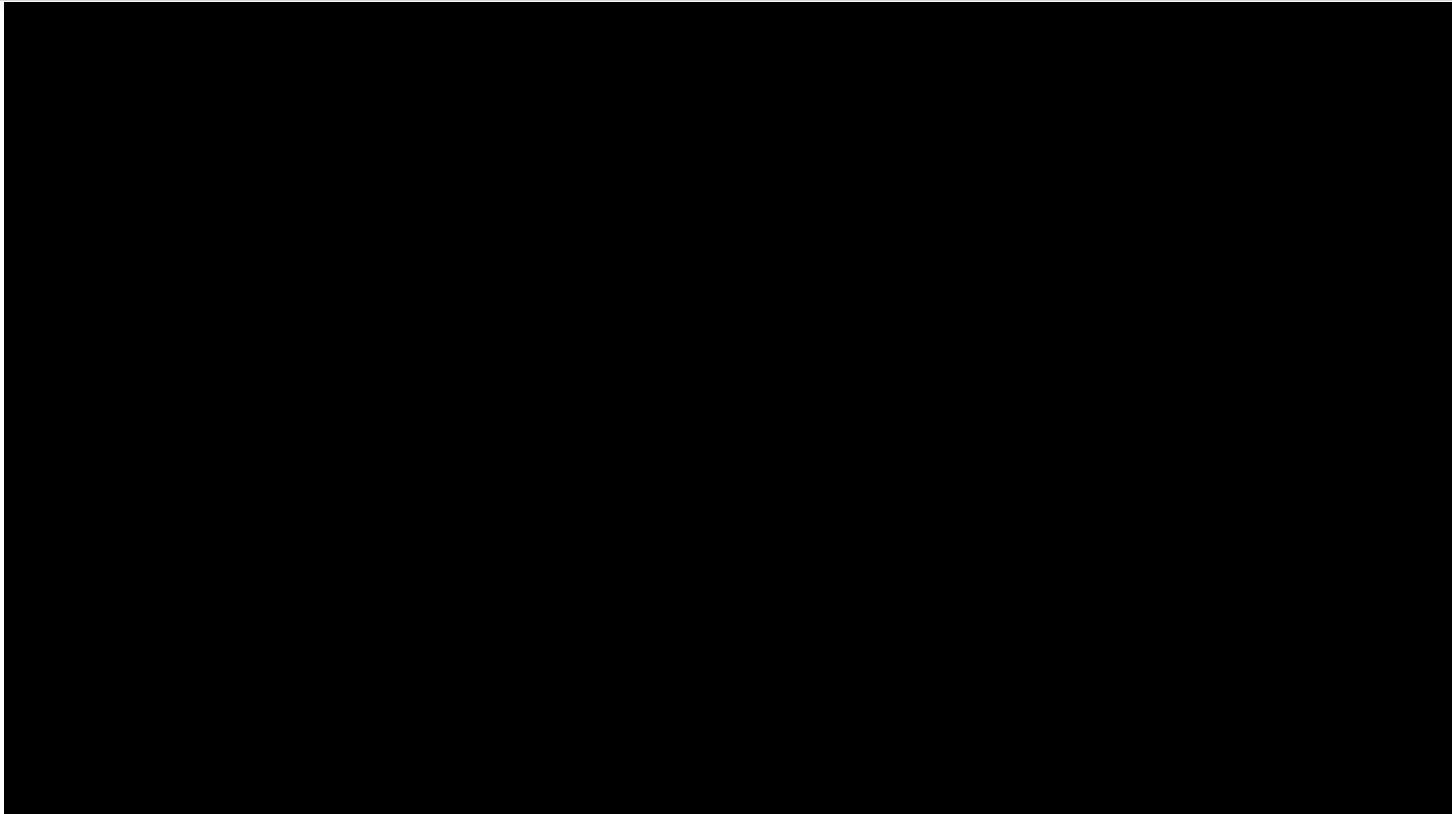
What is Pair Programming?



Pair programming is an **agile** software development technique in which two **programmers** work together at one workstation.

Pair Programming is proposed as one of the core practices of XP in the book

Pair Programming





AFS

Agile Full Stack
Developer Bootcamp
Thoughtworks @

Why Pair Programming?

/thoughtworks

Why Pair Programming?



More focused



Improve efficiency



Fewer defects



Knowledge transfer/
sharing



Enhance friendship



AFS

Agile Full Stack
Developer Bootcamp
Thoughtworks @

How to do Pair Programming?

/thoughtworks

Two Pair Programming styles

A large blue circle containing the text "Ping-Pong".

Ping-Pong

A large blue circle containing the text "Navigator-Driver".

Navigator-Driver

Some Reminders

Communicate
more

Change roles
regularly

Be open
minded

Confirm the
tasks first

Show your
gratitude

Give
feedback to
your pair



AFS

Agile Full Stack
Developer Bootcamp
Thoughtworks @

Spring Boot

/thoughtworks



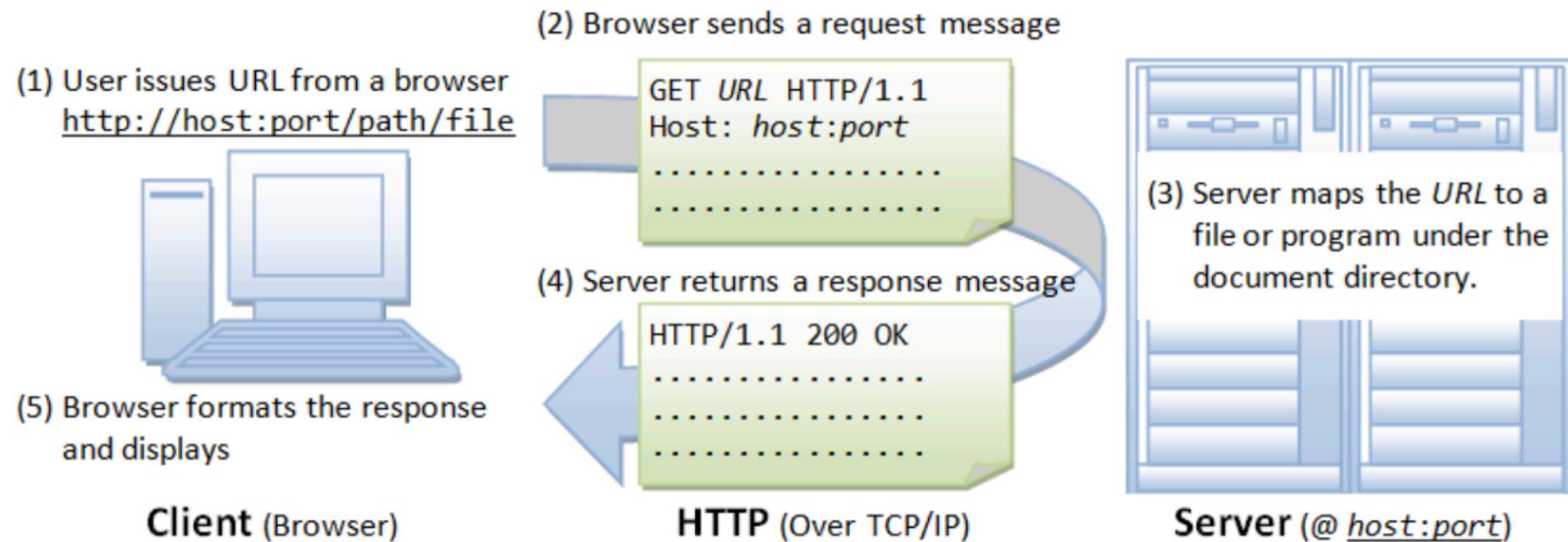
AFS

Agile Full Stack
Developer Bootcamp
Thoughtworks @

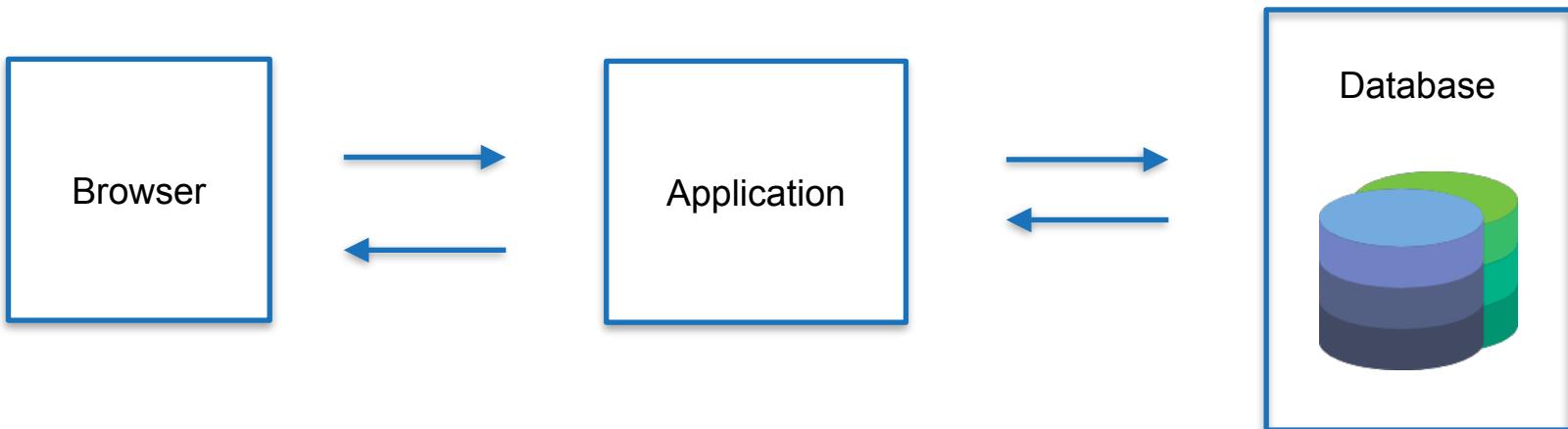
Spring Boot 101

/thoughtworks

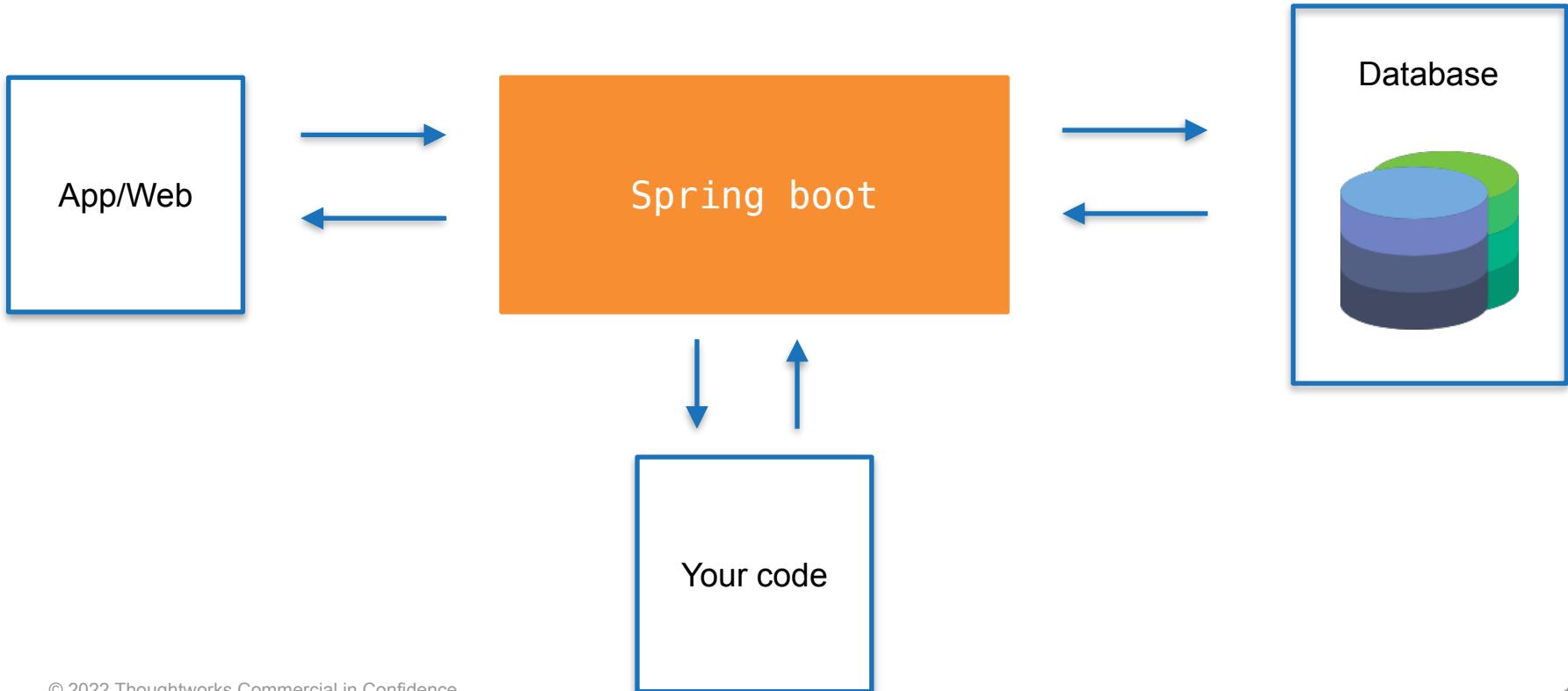
Web Development



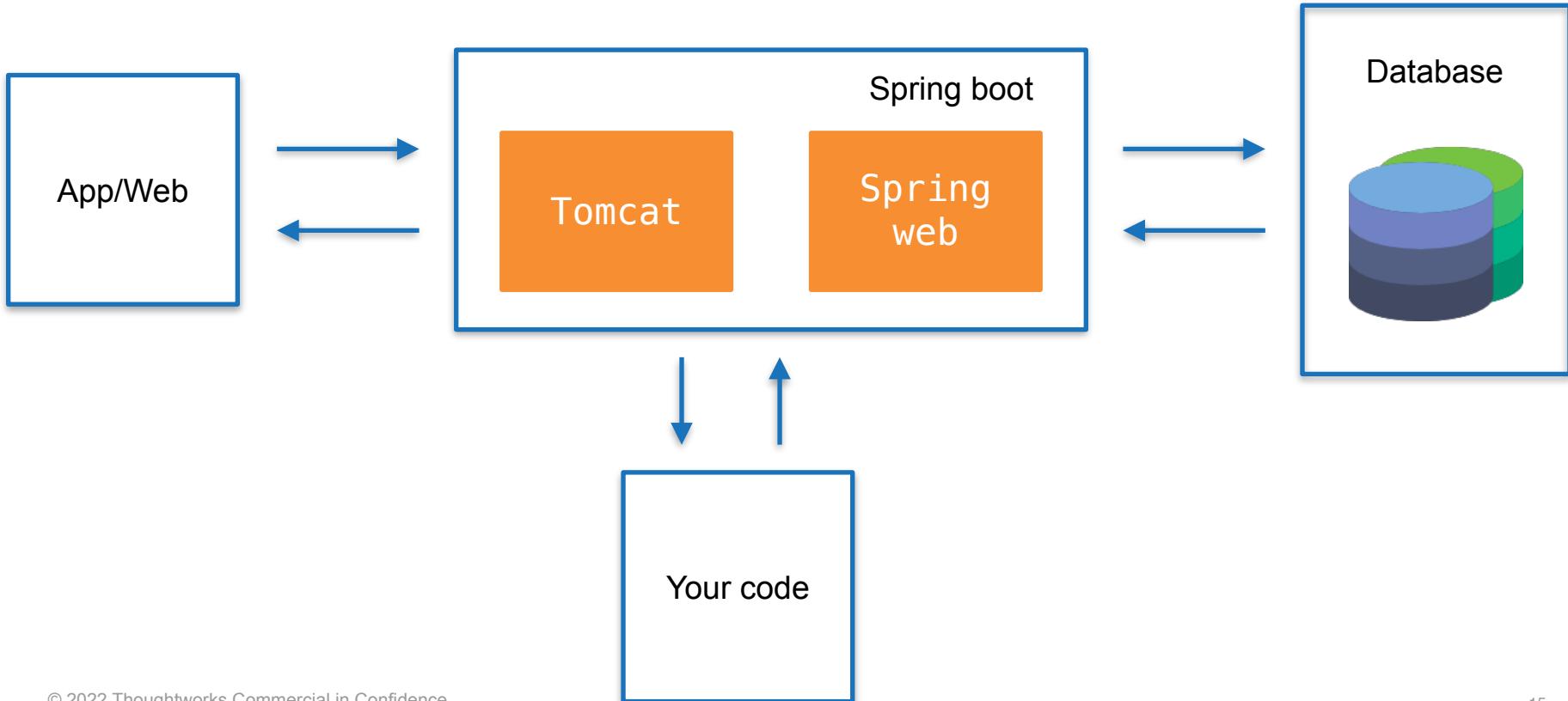
Web Development



Spring Boot is a box



Spring Boot is a box



Annotation is like the tag



@SpringBootApplication

@RestController

@RequestMapping

.....

Spring Boot

<https://start.spring.io/>

Spring Boot

Code Demo



AFS

Agile Full Stack
Developer Bootcamp
Thoughtworks @

Restful API Practice

/thoughtworks

Employees API Practice

Pair Programming

Preparation: select one repo and add partner as collaborator

Round 1: Kick-off

Partner A download the repo and implements the 1st API and push.
Partner B plays as Observer and provide feedback.

Round 2: Take Turns

Partner B pull the latest change and implements the 2nd API and push.
Partner A plays as Observer and provide feedback.

Round 3: Take Turns

Partner A pull the latest change and implements the 3rd API and push.
Partner B plays as Observer and provide feedback.

...

After the class, Partner A sends the copy of the code to B and finish the homework separately.

Employees API demo II

```
GET      /employees                      # get employee list
GET      /employees/1                     # get a specific employee by ID
GET      /employees?gender=male          # get all male employees

POST     /employees                      # add an employee
PUT      /employees/1                   response status 201 created
DELETE   /employees/1                   # update an employee age and salary
DELETE   /employees/1                   # delete an employee
                                                response status 204 no content

GET      /employees?page=1&pageSize=5    # Page query, page equals 1, pageSize equals 5
                                            {
                                                "id": 5,
                                                "name": "Lily",
                                                "age": 20,
                                                "gender": "Female",
                                                "salary": 8000
                                            }

Employee:
```

Company API

```
GET      /companies    #obtain company list with response of id, name  
GET      /companies/1  #obtain a certain specific company with  
response of id, name  
GET      /companies/1/employees # obtain list of all employee under  
a certain specific company  
GET      /companies?page=1&pageSize=5 #Page query, if page equals 1,  
pageSize equals 5, it will return the data in company list from index  
0 to index 4.  
POST     /companies    #add a company  
PUT      /companies/1  #update a company name  
DELETE   /companies/1  # delete a company
```

Company:

```
/companies/1
{
  "id": 1,
  "name": "spring"
}  
  
/company/1/employees
[
  {
    "id": 5,
    "name": "Lily",
    "age": 20,
    "gender": "Female",
    "salary": 8000
},
...  
]
```