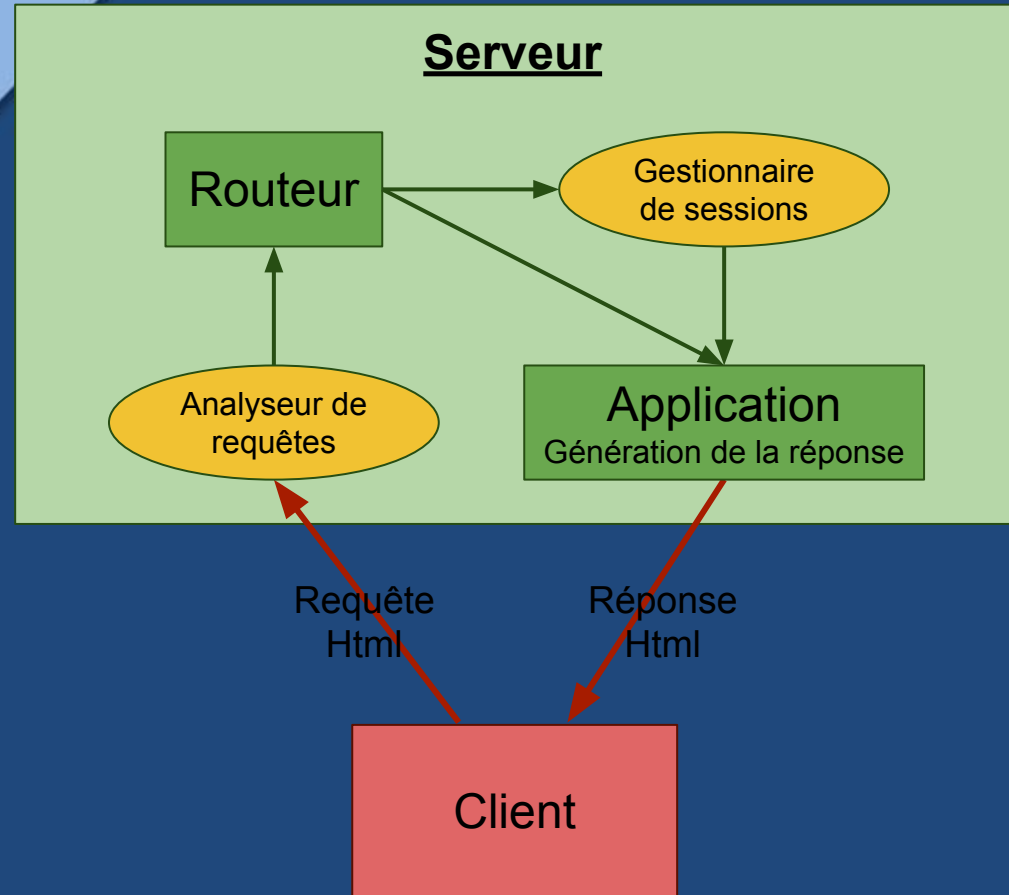


Développement d'Application Réticulaire

Réalisation d'un framework web

Manyanda CHITIMBO
Jérôme RAHAULT

Serveur de requêtes HTTP



Serveur de requêtes HTTP

Architecture

Connection

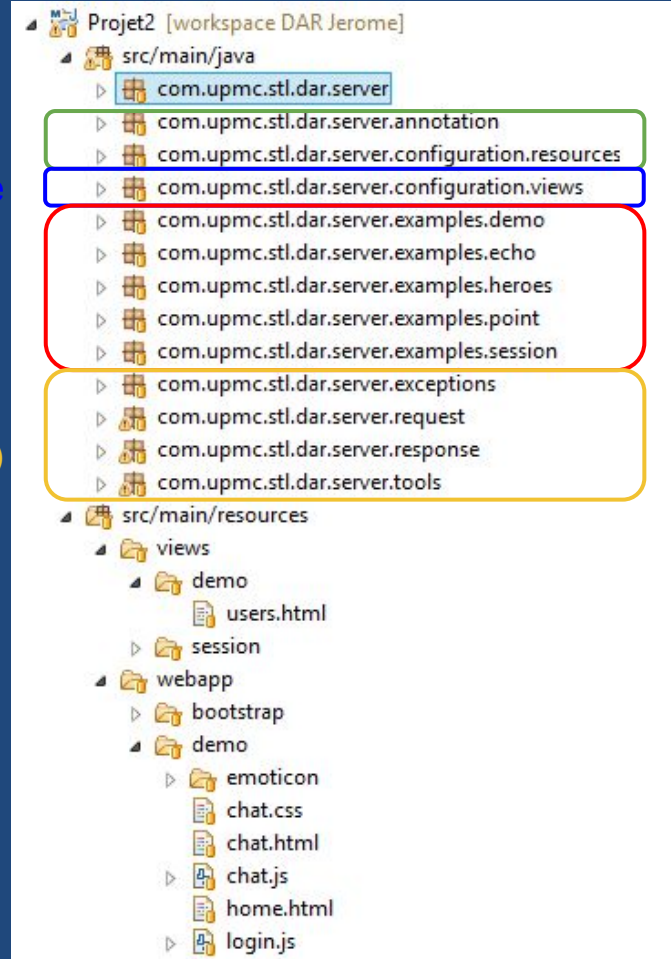
Routeur

Gestionnaire de template

Applications

Outils
(Gestionnaire de session..)

Partie Client



Serveur de requêtes HTTP

Analyseur de requêtes

GET /echo/html HTTP/1.1

Accept: */*

Connection: keep-alive

User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64)

AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.

0.2623.87 Safari/537.36

Host: localhost:30000

Accept-Encoding: gzip, deflate, sdch

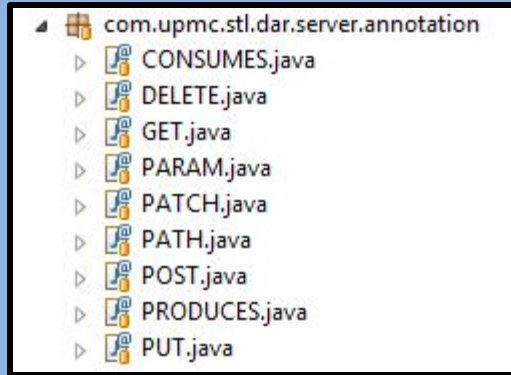
Accept-Language: fr-FR,fr;q=0.8,en-US;q=0.6,en;q=0.4

Content-Type: text/html



METHOD	URL	HEADERS	COOKIES
GET	/echo/html	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8 Upgrade-Insecure-Requests: 1 Connection: keep-alive User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.87 Safari/537.36 Host: localhost:30000 Accept-Encoding: gzip, deflate, sdch Accept-Language: fr-FR,fr;q=0.8,en-US;q=0.6,en;q=0.4	Cookie: LSID=DQAAAKâ€¦Eaem_vYg Cookie: HELLO=DQAAghjsgfhgsjhdgEaem_vYg

Routeur de requêtes



- Annotations Java
- Gestion des paramètres d'URL
- Gestion des conflits d'URL
- Invocation de ressource

Routeur de requêtes

Annotations

GET /point/425/y

```
@PATH("/point")
public class PointApplication {
    @GET
    @PRODUCES (ContentType.JSON)
    @PATH("/{<id>/y")
    public Integer getJsonY(@PARAM("<id>") Long id) {
        return points.get(id).getY();
    }
```

PUT /point/put/425?x=12&y=6

```
@PUT
@PATH("/put/{<id>")
    public void putPoint(@PARAM("<id>") Long id, UrlParameters params){
        points.put(id, new Point(
            Integer.valueOf(params.get("x")), Integer.valueOf(
                params.get("y"))));
    }
}
```

Gestionnaire de sessions

- Cookies
- Session
 - ❑ Un Cookie dont:
 - ❑ La clé est: ____sessionID
 - ❑ Sa valeur une unique de chaîne de taille 32.
 - ❑ Durée de 30 mins. Mais il est mis à jour à chaque requête.

Système de templates

- HTML classique en ajoutant le tag <render>
- Possible de structurer la view avec if ou for.
- Parsing du fichier vers un objet Vue.
- Générer une réponse html à chaque requête.

```
public ModelAndView goToChatPage() {  
    ModelAndView modelAndView = new ModelAndView("/views/demo/chat.html");  
    ArrayList<String> usernames = new ArrayList<String>();  
    for (User usr : users) {  
        usernames.add(usr.getUsername());  
    }  
    modelAndView.put("userList", usernames);  
    return modelAndView;  
}
```

```
<div class="list">  
    <render for = "user in userList">  
        {{user}}<br>  
    </render>  
</div>
```

Jean
Robert
DarkAngel64
BoGossDu75

Conclusion

Améliorations :

- Sécurité
- Couche de filtrage
- Stockage des sessions
- Système de templates élaboré

Application de
démonstration

Messagerie Instantanée

