

# Filtres

Les filtres permettent d'intercepter des requêtes HTTP avant qu'elles arrivent à leur Servlet afin d'exécuter de la logique. Cela est notamment utile pour restreindre l'accès à certaines routes à des utilisateurs non authentifiés ou ne possédant pas le bon rôle.

## Créer un filtre

Pour créer un filtre, il faut créer une classe qui implémente l'interface `Filter` et annotées avec `@WebFilter`. En paramètre de l'annotation, il faut passer le paramètre `urlPatterns`, pour préciser les routes qui seront filtrées par ce filtre.

```
@WebFilter(urlPatterns = "/admin/*")
public class AdminFilter implements Filter {

    public void doFilter(ServletRequest req, ServletResponse resp, FilterChain chain) throws
    ServletException, IOException {

    }

    public void init(FilterConfig config) throws ServletException {

    }

    public void destroy() {

    }

}
```

Par exemple ce filtre va intercepter les requêtes sur toutes les routes qui commencent par `"/admin/"`.

## Actions dans un filtre

Dans un filtre on peut récupérer la requête et la réponse HTTP par un simple cast :

```
var request = (HttpServletRequest) req;  
var response = (HttpServletResponse) resp;
```

On peut ensuite accéder à la session pour vérifier des informations ou faire une redirection. Pour autoriser la poursuite de la requête normalement utilisez :

```
chain.doFilter();
```

Et pour retourner un code HTTP particulier (401 Unauthorized ou 403 Forbidden) et arrêter là la requête :

```
response.setStatus(HttpServletResponse.SC_UNAUTHORIZED);  
return;
```

---

Created 2024-11-21 15:14:28 UTC by Nicolas  
Updated 2024-11-21 15:15:12 UTC by Nicolas