## Лабораторная работа №7

# Обеспечение безопасности. Фильтры

**Цель:**

Изучить механизмы обеспечения безопасности в веб-приложениях на платформе Java EE и получить навыки разработки и реализации политики безопасности для веб-приложения. Изучить механизм фильтрации запросов и получить навыки использования фильтров при разработке защищенных веб-приложений.

**Задание:**

1. Разработать и реализовать политику безопасности для веб-приложения, включающую не менее 3 ограничений безопасности и использующую не менее 2 ролей.
2. Использовать для веб-приложения домен безопасности на основе реляционной базы данных.
3. Опробовать следующие механизмы аутентификации: HTTP Basic и на базе форм.
4. Использовать методы программного ограничения доступа в сервлете или JSP-странице.
5. Разработать Java-класс для представления пользователей веб-приложения и использовать его для хранения и вывода сведений о текущем пользователе.
6. Реализовать фильтр, сохраняющий в сессии объект пользователя.

**Порядок выполнения работы**

1. Разработайте политику безопасности для веб-приложения. В текущей работе рекомендуется продолжить работу с проектом веб-приложения, созданным в ходе предыдущих лабораторных работ. Чтобы упростить настройку ограничений доступа в установочном дескрипторе можно определить ограничения доступа в виде следующей таблицы (в предположении, что не используются ограничения на среду передачи запросов):

|  |  |  |
| --- | --- | --- |
| Шаблон URL | Методы HTTP-запросов | Роли |
| \*.jsp | GET, POST, HEAD | никто |
| /UIController | GET, POST | user, admin |
| … | … | … |

1. Добавьте роли пользователей веб-приложения в установочный дескриптор. В редакторе установочного дескриптора выберите вкладку **Security**, раскройте пункт **Security Roles** и нажмите кнопку **Add…**. В диалоге добавления роли (рис. 1) введите название роли в поле **Role Name** и нажмите кнопку **OK**. Не используйте пробелы в названии роли.

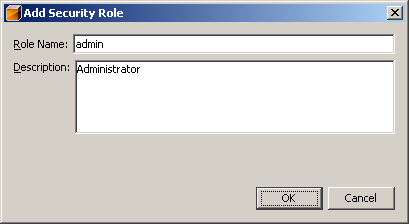


Рис. 1. Добавление роли

1. Добавьте ограничения безопасности в установочный дескриптор.
   1. В редакторе установочного дескриптора на вкладке **Security** в разделе **Security Constraints** нажмите кнопку **Add Security Constraint**. При этом в раздел добавляется пустая форма для ввода ограничения безопасности (на рис. 2 приведен вид уже заполненной формы). Введите в поле **Display Name** наименование ограничения.

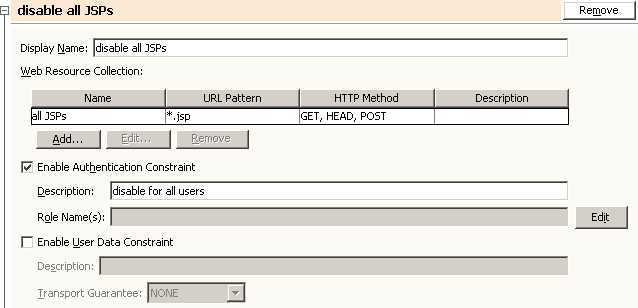


Рис. 2. Редактирование ограничения безопасности

* 1. Определите коллекцию ресурсов, на которые будет распространяться ограничение безопасности. Нажмите кнопку **Add…** в таблице **Web Resource Collection**. В диалоге добавления ресурса (рис. 3) укажите название ресурса (поле **Resource Name**). В соответствии с таблицей ограничений доступа укажите шаблон URL ресурса (поле **URL Pattern(s)**) и выберите методы HTTP-запросов (группа кнопок **HTTP Method(s)**), нажмите кнопку **OK**.
  2. Укажите роли пользователей, имеющих доступ к ресурсам. Установите флажок **Enable Authentication Constraint**, введите описание ролей в поле **Description**. Нажмите кнопку **Edit** справа от поля **Role Name(s)** для добавления ролей в ограничение безопасности. В диалоге выбора ролей (рис. 4) выберите нужные роли и нажмите кнопку Add >. Сверьте правый список ролей с таблицей ограничений доступа и нажмите кнопку **OK**.

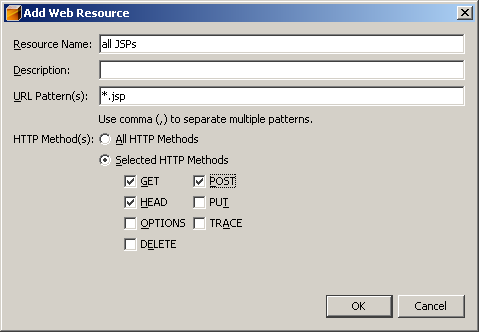


Рис. 3. Добавление ресурса в коллекцию

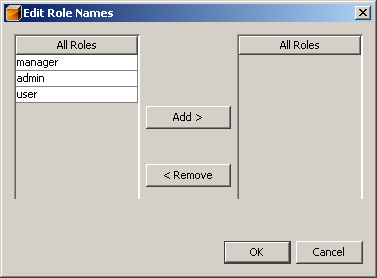


Рис. 4. Выбор ролей для ограничения безопасности

1. Создайте в MySQL таблицы для хранения учетных записей пользователей и ролей пользователей. Можно обойтись одной таблицей, если разработанная политика безопасности не предусматривает совмещения ролей. Так как таблица с учетными записями будет в дальнейшем использоваться также для хранения дополнительных сведений о пользователе (например, ФИО, дата рождения и проч.), то рекомендуется сразу создать все необходимые столбцы. Пример операторов создания таблиц для хранения учетных записей и ролей пользователей:

CREATE TABLE User (

username varchar(10) NOT NULL,

password varchar(20) NOT NULL,

fio varchar(100) NOT NULL,

birthdate date NOT NULL,

PRIMARY KEY (username)

);

CREATE TABLE User\_role (

username varchar(10) NOT NULL,

role varchar(20) NOT NULL,

PRIMARY KEY (username, role),

CONSTRAINT `FK\_User\_role\_1` FOREIGN KEY (username)

REFERENCES myuser (username)

);

1. Добавьте в базу данных несколько пользователей вместе с ролями.
2. Определите домен безопасности на основе созданной базы данных для использования в рамках веб-приложения. Указанный контекст использования домена безопасности позволяет выполнить его конфигурирование в файле context.xml, определяющем настройки веб-приложения, специфичные для веб-сервера Tomcat. Кроме того, можно определить домен безопасности для всего веб-сервера или для отдельного виртуального хоста.
   1. Откройте файл context.xml, расположенный в папке **Configuration Files** проекта веб-приложения и добавьте внутрь элемента Context элемент Realm с описанием домена безопасности. Описание атрибутов элемента Realm приведено в таблице ниже. Пример описания домена безопасности для таблиц, приведенных выше:

<Realm className="org.apache.catalina.realm.DataSourceRealm" debug="99"

dataSourceName="jdbc/clients"

userTable="myuser" userNameCol="username" userCredCol="password"

userRoleTable="myuser\_role" roleNameCol="role"/>

|  |  |
| --- | --- |
| Атрибут | Описание |
| className | Имя класса, реализующего домен безопасности.  Для домена безопасности, использующего в качестве механизма хранения реляционную базу данных с подключением через источник данных, должно быть равно  org.apache.catalina.realm.DataSourceRealm |
| debug | Уровень отладочных сообщений (99 - наиболее детальные сообщения). |
| dataSourceName | JNDI-имя источника данных. |
| userTable | Таблица учетных записей пользователей. |
| userNameCol | Столбец, содержащий имя пользователя. Название данного столбца в таблицах учетных записей и ролей должно совпадать. |
| userCredCol | Столбец, содержащий пароль пользователя. |
| userRoleTable | Таблица ролей пользователей. |
| roleNameCol | Столбец, содержащий название роли пользователя. |

* 1. Сохраните файл context.xml. Установите веб-приложение (пункт **Deploy Project** в контекстном меню проекта веб-приложения).
  2. Запустите утилиту администрирования веб-сервера Tomcat и убедитесь, что в контексте веб-приложения появился домен безопасности (пример для веб-приложения Clients приведен на рис. 5).

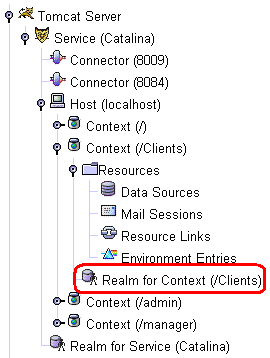


Рис. 5. Домен безопасности для веб-приложения

1. Установите для аутентификации пользователей механизм HTTP Basic. В редакторе установочного дескриптора выберите вкладку **Security**, раскройте пункт **Login Configuration**, выберите переключатель **Basic** и введите произвольную строку с наименованием домена в поле **Realm Name** (рис. 6).

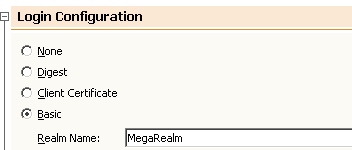


Рис. 6. Домен безопасности для веб-приложения

1. Запустите веб-приложение и перейдите на какую-либо защищенную страницу. При этом браузер должен вывести окно аутентификации в указанном вами домене (рис. 7). Попробуйте вводить правильные и неправильные комбинации имени пользователя и пароля.

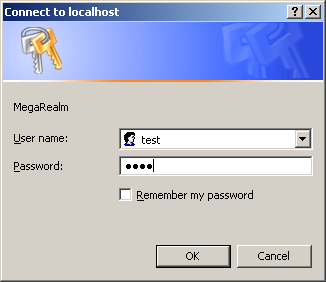


Рис. 7. Окно аутентификации браузера Internet Explorer

1. Установите для аутентификации пользователей механизм с использованием формы.
   1. Создайте JSP-страницу с формой для аутентификации пользователя, отвечающей требованиям спецификации сервлетов.
   2. Создайте JSP-страницу, которая будет выводиться в случае ошибки аутентификации. Данная JSP-страница может включать страницу с аутентификационной формой, чтобы пользователь имел возможность повторить процедуру аутентификации.
   3. В редакторе установочного дескриптора выберите вкладку **Security**, раскройте пункт **Login Configuration** и выберите переключатель **Form**. Установите путь к странице с формой аутентификации в поле **Form Login Page** и путь к странице ошибки аутентификации в поле **Form Error Page** (рис. 8). При этом можно воспользоваться кнопками **Browse…**, которые выводят диалог выбора файлов в папке Web Pages.



Рис. 8. Настройка аутентификации с использованием формы

1. Запустите веб-приложение и перейдите на какую-либо защищенную страницу. При этом должна открыться созданная ранее страница с формой аутентификации пользователя. Попробуйте вводить правильные и неправильные комбинации имени пользователя и пароля.
2. Измените существующую JSP-страницу или сервлет таким образом, чтобы в ней (в нем) использовались методы программного ограничения доступа. Например, можно выводить в результат имя пользователя, либо выводить некоторые данные, только если пользователь выполняет определенную роль.
3. Разработайте Java-класс для представления пользователя веб-приложения и реализуйте в DAO-классе метод для получения пользователя по имени учетной записи.
4. Реализуйте фильтр, сохраняющий в сессии объект пользователя.
   1. Создайте в проекте новый фильтр (файл типа **Web / Filter**). На первом шаге мастера создания фильтра (рис. 9) установите имя класса фильтра (поле **Class Name**) и нажмите кнопку **Next >**. На втором шаге мастера (рис. 10) можно согласиться со всеми установками, произведенными по умолчанию и нажать кнопку **Next >**. При этом в установочный дескриптор будут добавлены сведения о фильтре и об обрабатываемых им запросах. На третьем шаге мастера (рис. 11) можно добавить параметры инициализации фильтра и нажать кнопку **Finish** для окончания создания фильтра.

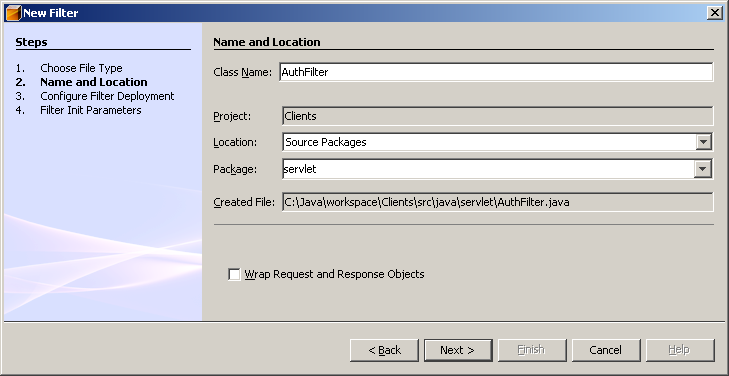


Рис. 9. Мастер создания фильтра: шаг 1

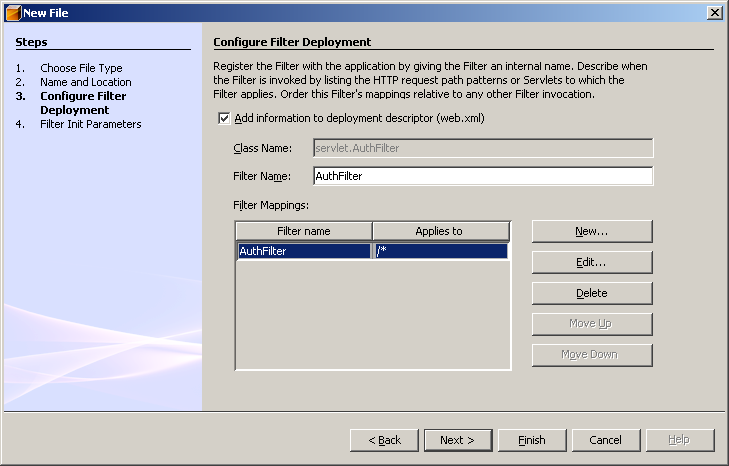


Рис. 10. Мастер создания фильтра: шаг 2

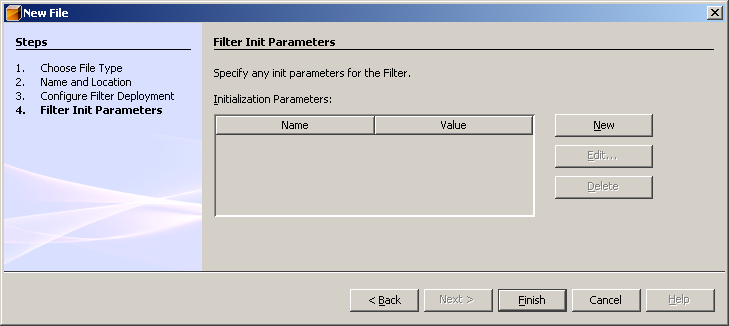


Рис. 11. Мастер создания фильтра: шаг 3

* 1. В редакторе установочного дескриптора выберите вкладку **Filters** и убедитесь, что фильтр зарегистрирован и в таблице **Filter Mappings** для него определена обработка всех запросов к веб-приложению (рис. 12).

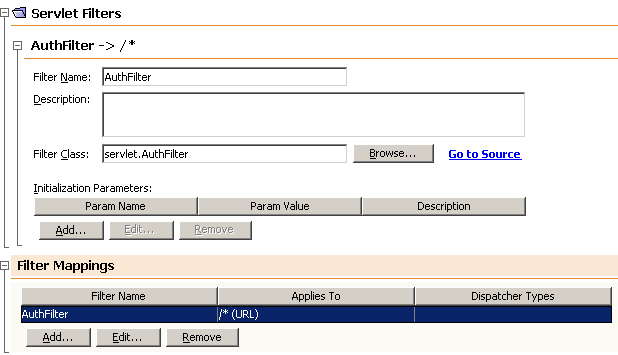


Рис. 12. Настройки фильтра в установочном дескрипторе

* 1. Изучите исходный код фильтра, сгенерированный мастером.
  2. Добавьте в метод doBeforeProcessing() код, добавляющий в сессию объект пользователя (см. конспект лекции, посвященной безопасности веб-приложений).
  3. Добавьте в незащищенную JSP-страницу вывод информации из объекта пользователя, например, ФИО или даты рождения.
  4. Проверьте правильность функционирования фильтра, обращаясь к данной странице до и после аутентификации.