Фильтры

|  |  |  |  |
| --- | --- | --- | --- |
| № урока: | 5 | Курс: | AngularJS |
|  | | |  |
| Средства обучения: | | | Visual Studio 2013 Web Express или коммерческая версия |

# Обзор, цель и назначение урока

Научится использовать встроенные фильтры AngularJS, понимать ключевые отличия между применением функций для фильтрации данных и применением фильтров, создавать пользовательские фильтры и познакомится с сервисом $filter.

**Изучив материал данного занятия, учащийся сможет:**

* Создавать и применять фильтры в AngularJS.
* Использовать цепочки фильтров.
* Применять функции для сортировки данных с помощью фильтров.
* Использовать фильтры для работы с единичными данными и для работы с коллекциями данных.
* Использовать встроенный сервис AngularJS - $filter.

# Содержание урока

1. Работа со встроенными фильтрами.
2. Фильтрация массивов объектов.
3. Локализация и форматирование отфильтрованных данных.
4. Создание «цепочек» фильтров.
5. Создание пользовательских фильтров.
6. Создание фильтров, использующих другие фильтры.

# Резюме

* Фильтры предназначены для преобразования данных перед тем как их обработает директива и перед их отображением во View без модификации данных в scope. С помощью фильтров можно отобразить данные из одного источника в нескольких местах приложения (с применением фильтра и без). Фильтры могут выполнять множество функций, но в большинстве случаев они используются для форматирования и сортировки данных.
* В AngularJS существует несколько типов фильтров: для обработки единичных данных и для работы с коллекциями данных.

Встроенные фильтры для работы с единичными данными:

* currency – используется для форматирования валюты
* date – используется для форматирования даты
* json – генерирует JSON строку из JavaScript объекта
* number – используется для форматирования числовых значений
* uppercase – преобразует строку в верхний регистр
* lowercase – преобразует строку в нижний регистр
* Так же в AngularJS существует возможность форматирования с помощью функций, как и в обычном JavaScript. Но такой подход имеет ряд недостатков:
* форматирование задается жестко в теле функции и его далее сложно изменить. В фильтрах его можно менять в зависимости от культуры;
* данные модифицируются непосредственно в источнике;
* теряется возможность использовать одни и те же данные повторно во View так как они уже модифицированы.
* В AngularJS присутствует встроенная локализация для форматирования данных.

По умолчанию используется текущая культура которая установлена на машине, но существует возможность применять специальные файлы для локализации под разные культуры. Эти файлы находятся по адресу: <https://code.angularjs.org/1.2.23/i18n/>

Обратите внимание что данная локализация распространяется лишь на небольшой перечень свойств и для полноценной локализации её не достаточно.

* Фильтры для форматирования коллекций данных:
* limitTo – используется для ограничения отображаемых элементов из источника данных, так же существует возможность применения этого фильтра к строке, где он каждый символ обрабатывает как элемент массива;
* orderBy – один из наиболее широко используемых фильтров, предназначен для сортировки элементов массива. Для сортировки можно указывать как функцию, так и свойство, при этом свойство следует указывать в одинарных кавычках.
* В AngularJS существует возможность использования цепочек фильтров:

“Item in items |orderBy: ’+price’ | limitTo: 3 “

В таком случае каждому следующему фильтру передаются данные уже обработанные предыдущим фильтром.

* Для того чтобы создать пользовательский фильтр нужно использовать метод filter который есть у модуля: Module.filter(). Этот метод принимает 2 аргумента:
* 1) Имя фильтра;
* 2) Factory function которая создает worker function.

Далее его можно использовать в разметке как обычный встроенный фильтр.

* Сервис $filter используется, когда вам нужно в коде приложения вызывать существующие фильтры, например, для объединения функционала, создания оберток или для обработки данных непосредственно в функции.

Для работы с этим сервисом его нужно просто указать в аргументах функции и AngularJS используя Dependency Injection передаст $filter в качестве аргумента функции.

# Закрепление материала

* Объясните работу фильтра currency. Как можно задать необходимую валюту?
* Перечислите недостатки фильтрации с помощью функций и преимущества фильтрации с помощью фильтров.
* Назовите особенности применения фильтра orderBy.
* Как создать пользовательский фильтр?
* Можно ли в коде приложения вызывать встроенный фильтры? Опишите.

# Дополнительное задание

Задание

Создайте приложение – дневник в котором пользователь может вносить записи, возле каждой его записи указана дата, и записи отсортированы по дате.

# Самостоятельная деятельность учащегося

Задание 1

Расширьте предыдущий пример добавив возможность пользователю выбирать способ сортировки (по дате, количество отображаемых записей). Добавьте локализацию с помощью файла и предоставьте пользователю выбирать культуру для форматирования (с помощью выпадающего списка).

Задание 2

Создайте пользовательский фильтр который можно применять для сортировки массива чисел.

Фильтр должен в начало массива располагать по порядку все четные значения, далее по порядку все не четные значения.

# Рекомендуемые ресурсы

Filters:

<https://docs.angularjs.org/api/ng/filter>

$filter:

<https://docs.angularjs.org/api/ng/service/$filter>

Culture files:

<https://code.angularjs.org/1.2.23/i18n/>