Грамматики запросов

Грамматики (модули) описывают правила, по которым фразы пользователя воспринимаются вашей программой. А именно - какие команды и в каком контексте в каждый момент времени могут быть выполнены.

Общие сведения

Грамматики - это обычные xml-файлы, хранящиеся в директориях вашего проекта.

Где создаются грамматики

Грамматики создаются в директории `xml` вашего проекта. Если необходимо описать грамматики для различных языков, то необходимо воспользоваться стандартной процедурой локализации ресурсов в Android-е - т.е. создать файлы грамматик в директориях вида `xml-<код языка>`.

Что содержится в грамматиках

Грамматики содержат список команд и паттернов, используемых вашим приложением. В них не содержится код, выполняющий какую-либо логику. Грамматики могут использовать паттерны, определенные в других грамматиках (например, доступные всем паттерны чисел или времени).

Структура грамматик

```
<?xml version="1.0" encoding="utf-8"?>
<module xmlns:android="http://schemas.android.com/apk/res/android">
    <uses-pattern name="..."/>

    <pattern name="..." value="..."/>
    <pattern name="..." uri="..."/>
```

```
    <pattern name="...">
        <pattern value="..."/>
    </pattern>

    <command android:id="@+id/...">
        <pattern value="..."/>

        <command android:id="@+id/...">
            <pattern value="..."/>
        </command>
    </command>
</module>
```

Корневым элементом является тэг `module`. В нём указывается схема `android`.

Модуль может содержать неограниченное количество паттернов и команд.

pattern

Описывает шаблон (паттерн) для пользовательской фразы. Синтаксис паттернов детально описан здесь.

Паттерны, указанные непосредственно под тэгом `module` - это общие паттерны, которые используются в других паттернах. Паттерны под тэгами `command` - это непосредственно те паттерны, которые обрабатывают запросы для каждой конкретной команды.

Атрибут `name`

У каждого паттерна может быть имя (атрибут `name`), которое однозначно идентифицирует паттерн в модуле, и по которому можно ссылаться на него в других паттернах. Также это имя будет указано в семантическом дереве, которое будет передано в запросе к вашему приложению.

Атрибут `value`

Паттерн может быть задан явно с помощью синтаксиса паттернов в атрибуте `value`. Либо неявно с использованием атрибута `uri`.

Атрибут `uri`

Здесь указывается URI, по которому будет сформирован паттерн. Это либо URI локального контент-провайдера со схемой `content://`, либо удалённый паттерн (поддерживаются схемы `http://` и `ws://`).

Данное руководство пока не описывает возможность использовать удалённые паттерны.

Паттерн может быть задан либо явно с помощью `value`, либо неявно с помощью `uri`. Использовать оба варианта одновременно для одного паттерна нельзя!

Вложенность паттернов

Паттерны могут быть вложенными. Это означает более наглядную запись альтернатив. Например, паттерн вида

```
    <pattern name="Pattern">
        <pattern value="one"/>
        <pattern value="two"/>
    </pattern>
```

Равнозначен паттерну вида

```
    <pattern name="Pattern" value="(one|two)"/>
```

Еще одной особенностью такой вложенности является возможность задать каждому вложенному паттерну имя.

uses-pattern

Этот тэг импортирует паттерн, определённый в другом модуле.

Атрибут `name`

Здесь указывается уникальное имя паттерна, который необходимо импортировать.

Список глобальных паттернов

Существует набор глобальных (общих) паттернов, доступных всем приложениям.

Number - числа в различных формах. Этому паттерну соответствует конвертер *NumberConverter* для извлечения целого числа.

Date - даты. Конвертер DateConverter возвращает обёртку Date.

Time - время. Конвертер TimeConverter возвращает обёртку Time.

DateTime - дата-время. Используется когда во фразе необходимо наличие неразделимой последовательности даты и времени. Конвертер DateTimeConverter возвращает обёртку DateTime.

Location - географическое месторасположение. Как абсолютные (города), так и относительные (здесь, там и т.п.). Ковертер LocationConverter позволяет получить местоположение в виде обёртки Location.

Contact - контакт из контактной книги пользователя. ContactConverter возвращает обёртку Contact.

Подробнее о конвертерах можно узнать здесь.

command

В этом тэге указывается идентификатор команды, которая будет передана агенту, обрабатывающему запросы к вашему приложению. В команду вложен по крайней мере один паттерн, а также может быть вложена другая команда.

Атрибут `android:id`

Здесь указывается идентификатор команды. Записывается в формате `@+id/<идентификатор>`, либо `@id/<идентификатор>`. Т.к. все xml-файлы компилируются, эти идентификаторы впоследствии будут доступны через `R.id`.

Вложенные паттерны

В каждой команде должен присутствовать хотя бы один паттерн - это фраза, на которую должна сработать команда. Другими словами, при

обработке текста от системы распознавания, ассистент выберет паттерн, наиболее точно подходящий под этот текст, и передаст управление соответствующему агенту с указанием команды, в которой сработал паттерн.

Вложенные команды

В команду верхнего уровня (ту, которая описана непосредственно в тэге `module`) можно вложить несколько дочерних команд. Эти команды будут доступны пользователю только после того, как сработает команда верхнего уровня, в которую они вложены (читайте подробнее в разделе Scopes).

Например, после фразы *"Расскажи погоду"* пользователь может сказать *"На завтра"* или *"В Москве"*, но без первоначальной команды эти фразы бессмысленны и соответственно не будут доступны пользователю.

Вложенные команды формируют автоматически-разворачиваемые области (scopes), формируя таким образом диалоговый контекст. Конечно, можно управлять этим процессом более гибко. Для этого нужно воспользоваться соответствующими методами API, где можно указывать модули (грамматики), которые сейчас должны обрабатывать запросы пользователя. Об этом более подробно описано в разделе Продвинутые функции.

sample

Этот тэг используется только внутри тэга `command`. В единственном атрибуте `value` нужно указать пример пользовательского запроса, который может использоваться для команды.

> Впоследствии это значение может использоваться в пользовательском интерфейсе *Ассистента на русском* для отображения примеров того, что сейчас можно сказать.