<data>

SYNTAX:

```
<data android:host="string"
      android:mimeType="string"
      android:path="string"
      android:pathPattern="string"
      android:pathPrefix="string"
      android:port="string"
      android:scheme="string" />
```

CONTAINED IN:

<intent-filter>

DESCRIPTION:

Adds a data specification to an intent filter. The specification can be just a data type (the mimeType attribute), just a URI, or both a data type and a URI. A URI is specified by separate attributes for each of its parts:

scheme://host:port/path *or* pathPrefix *or* pathPattern

These attributes are optional, but also mutually dependent: If a scheme is not specified for the intent filter, all the other URI attributes are ignored. If a host is not specified for the filter, the port attribute and all the path attributes are ignored.

All the <data> elements contained within the same <intent-filter> element contribute to the same filter. So, for example, the following filter specification,

```
<intent-filter . . . >
    <data android:scheme="something" android:host="project.example.com" />
    . . .
</intent-filter>
```

is equivalent to this one:

```
<intent-filter . . . >
    <data android:scheme="something" />
    <data android:host="project.example.com" />
    . . .
</intent-filter>
```

You can place any number of <data> elements inside an <intent-filter> to give it multiple data options. None of its attributes have default values.

Information on how intent filters work, including the rules for how Intent objects are matched against filters, can be found in another document, Intents and Intent Filters. See also the Intent Filters section in the introduction.

ATTRIBUTES:

android:host

The host part of a URI authority. This attribute is meaningless unless a scheme attribute is also specified for the filter.

> Note: host name matching in the Android framework is case-sensitive, unlike the formal RFC. As a

> result, you should always specify host names using lowercase letters.

`android:mimeType`
A MIME media type, such as `image/jpeg` or `audio/mpeg4-generic`. The subtype can be the asterisk wildcard (`*`) to indicate that any subtype matches.

> Note: MIME type matching in the Android framework is case-sensitive, unlike formal RFC MIME types. As a result, you should always specify MIME types using lowercase letters.

`android:path`
`android:pathPrefix`
`android:pathPattern`
The path part of a URI. The `path` attribute specifies a complete path that is matched against the complete path in an Intent object. The `pathPrefix` attribute specifies a partial path that is matched against only the initial part of the path in the Intent object. The `pathPattern` attribute specifies a complete path that is matched against the complete path in the Intent object, but it can contain the following wildcards:

- An asterisk ('`*`') matches a sequence of 0 to many occurrences of the immediately preceding character.
- A period followed by an asterisk ("`.*`") matches any sequence of 0 to many characters.

Because '`\`' is used as an escape character when the string is read from XML (before it is parsed as a pattern), you will need to double-escape: For example, a literal '`*`' would be written as "`\\*`" and a literal '`\`' would be written as "`\\\\`". This is basically the same as what you would need to write if constructing the string in Java code.

For more information on these three types of patterns, see the descriptions of `PATTERN_LITERAL`, `PATTERN_PREFIX`, and `PATTERN_SIMPLE_GLOB` in the `PatternMatcher` class.

These attributes are meaningful only if the `scheme` and `host` attributes are also specified for the filter.

`android:port`
The port part of a URI authority. This attribute is meaningful only if the `scheme` and `host` attributes are also specified for the filter.

`android:scheme`
The scheme part of a URI. This is the minimal essential attribute for specifying a URI; at least one `scheme` attribute must be set for the filter, or none of the other URI attributes are meaningful.

A scheme is specified without the trailing colon (for example, `http`, rather than `http:`).

If the filter has a data type set (the `mimeType` attribute) but no scheme, the `content:` and `file:` schemes are assumed.

> Note: scheme matching in the Android framework is case-sensitive, unlike the RFC. As a result, you should always specify schemes using lowercase letters.

INTRODUCED IN:
API Level 1

SEE ALSO:
`<action>`
`<category>`

← Back to The AndroidManifest.xml File ↑ Go to top