NativeSPRequestMapHost

The <Host> element is used to apply content rules to requests to a specific virtual host (or often to the two virtual hosts running on the default ports).

Attributes

Content Specifiers

- name (string)
  - Required attribute, specifies the hostname of requests to match against. This must correspond to the "canonical" name of the virtual host, or the client can circumvent the matching process.
- scheme ("http" or "https")
  - Optional scheme of requests to match against. Used to match only requests using a specific protocol. If omitted, any protocol will be matched.
- port (integer)
  - Optional port of requests to match against. Used to match only requests using a specific port. If omitted, any request to the default port for the protocol used (either 80 or 443) will be matched.

Content Settings

XML attributes corresponding to request mapper properties are used.

Child Elements

Access Control

One of the following elements can be used to attach an access control policy to the resource. This is a violation of the axiom that the SP doesn't do access control, but it's really just a call-out that has some predefined plugins you can use as examples to create more.

- <htaccess>
  - Enables Apache .htaccess support during the authorization phase. This is automatic and implicit for the "Native" request mapper, but can be enabled by hand if the "XML" request mapper is used. Note that this will fail for non-Apache servers.
- <AccessControlProvider>
  - Attaches a custom access control policy supported by a plugin.
- <AccessControl>
  - Attaches an access control policy using the sample XML-based plugin provided with the SP. This is just a short-hand for embedding the policy in the element above, if you want the policy inside the same file.

If no element is included (or inherited or implicitly enabled), any access control is left to the resource.

If an error occurs when processing this element, a dummy policy to deny access is installed to prevent accidental exposure.

Nested Content Specifiers

Zero or more of these "overrides" to match specific content on the virtual host can be included.

- <Path>
  - Matches requests whose first path component is an exact match for the element.
- <PathRegex>
  - Matches requests with a path that matches the element's expression.
- <Query>
  - Matches requests containing a query string parameter satisfying the element.

Matching is done within a <Host> element as follows:

1. First, by examining <Path> elements in order.
2. Then, by checking any <PathRegex> elements in order against the part of the path that was not matched in the first step.
3. Finally, by examining any <Query> elements in order.

Once a matching child element is found, the process steps "into" that element and no other siblings will be applied. Thus, siblings cannot overlap.

For more details on how the request mapping process works, see the request mapper HOWTO.