

# Project Configuration API

## Contents

- [Introduction](#)
- [Get A Listing Of Tools With Active Configurations At The Project Level](#)
- [Get A Config Object For A Specific Tool In Your Project](#)
- [Store A Project-specific Config Value For A Specific Tool](#)

## Introduction

The XNAT config service is a flexible API that allows individual features and plugins to store or retrieve configuration information as needed. Each feature or plugin that stores a unique config object is referred to as a "tool." These configurations can be stored on a site-wide or project level. Configurations can correspond to a control panel that allows users to directly edit settings, but the information stored in the config API may expand beyond what is displayed in the control panel. Since each tool is unique, this API documentation will only focus on the general functionality associated with storing and retrieving config settings.

For an example on how to use these API on a core XNAT config setting, see: [Using the XNAT API: How to Configure Scan Quality Labels](#).

## Get A Listing Of Tools With Active Configurations At The Project Level

```
GET - /data/projects/{projectId}/config
```

### Parameters:

<b>{projectId}</b>	Required path parameter
<b>format</b>	Optional querystring parameter. Specifies the output format. <ul style="list-style-type: none"><li>• json (default)</li><li>• html</li><li>• xml</li><li>• csv</li></ul>

### Response:

The response provides a list of tool IDs. Each tool ID can be used as a path parameter to retrieve specific config information about that tool.

```
{
  "ResultSet": {
    "Result": [
      {
        "tool": "string"
      },
      ...
    ]
  }
}
```

## Get A Config Object For A Specific Tool In Your Project

```
GET - /data/projects/{project-id}/config/{tool-id}
```

### Parameters:

<b>{project-id}</b>	Required path parameter
<b>{tool-id}</b>	Required path parameter – uses the same string as listed above in the listing of all config tools
<b>format</b>	Optional querystring parameter. Specifies the output format. <ul style="list-style-type: none"> <li>• json (default)</li> <li>• html</li> <li>• xml</li> <li>• csv</li> </ul>
<b>accept-not-found</b>	Optional querystring parameter. <ul style="list-style-type: none"> <li>• true: returns a 204 - Content Not Found code, if no configs have been set for this tool</li> </ul>

**Response Code:**

<b>200</b>	A project-specific config exists for this tool
<b>204</b>	A project-specific config has never been set for this tool (if accept-not-found = true)
<b>404</b>	A project-specific config has never been set for this tool

**Response Format:**

The inner format of the response for each tool will follow a common structure, but the "contents" parameter may contain a simple string or a complex JSON object, depending on the tool.

```
{
  "ResultSet": {
    "Result": [
      {
        "contents": "string",
        "create_date": "datetime",
        "path": "string",
        "reason": "string",
        "project": "string",
        "status": "string",
        "tool": "string",
        "unversioned": "boolean",
        "user": "string",
        "version": "integer"
      },
      ...
    ]
  }
}
```

Response Parameter	
contents	The actual contents of the configuration file
create_date	The date/time the configuration was created
path	the path to the configuration file
reason	The user provided reason for uploading the configuration
status	Enumerated setting: "enabled" / "disabled"
tool	The name of a tool that has configuration files
unversioned	Whether or not this config object is "versioned." <ul style="list-style-type: none"> <li>• If true, new configs will not overwrite old configs, and all versions of a config will be stored.</li> <li>• If false, new configs WILL overwrite old configs, and only the latest version of a config will be stored.</li> </ul>

user	The user who created the configuration
version	version number of the configuration contents

## Store A Project-specific Config Value For A Specific Tool

```
PUT - /data/projects/{project-id}/config/{tool-id}/{file-path}
```

### Parameters

<b>{project-id}</b>	Required path parameter
<b>{tool-id}</b>	Required path parameter – uses the "tool" string as listed above
<b>{file-path}</b>	Required path parameter – uses the "path" string as listed above
<b>inbody</b>	Required querystring parameter <ul style="list-style-type: none"> <li>"true" indicates that the params object is stored in the body of the PUT request.</li> </ul>
<b>contents</b>	Required parameter, either in body or querystring. The format of your config value should match the format expected by the tool. If you are not sure what that is, perform a GET to get the current value.

**Response:** No response object is returned. A "201" response code indicates that the config value was stored.