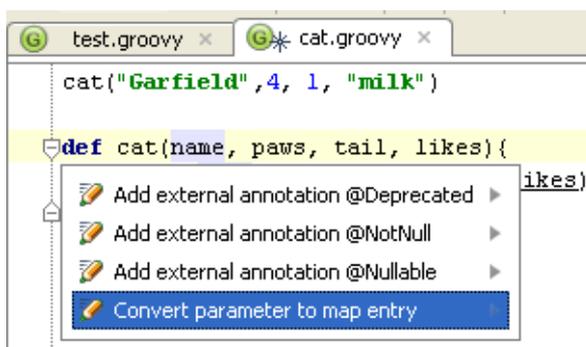# Working with Lists and Maps

In this section you can find some examples of working with lists and maps in IntelliJ IDEA:

- To convert a parameter of a function to a map entry
- To convert Groovy map to a class instance
- To produce a list or a map
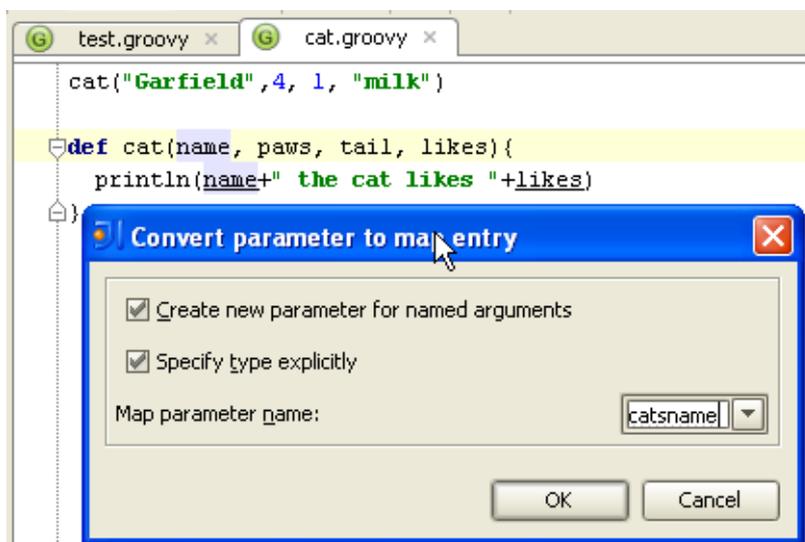- To inline a list or a map

To convert a parameter of a function to a map entry

If a function has a lengthy list of named arguments, you can reduce it by representing parameters as map entries. For this purpose, IntelliJ IDEA suggests the *Convert parameter to map entry* intention action.
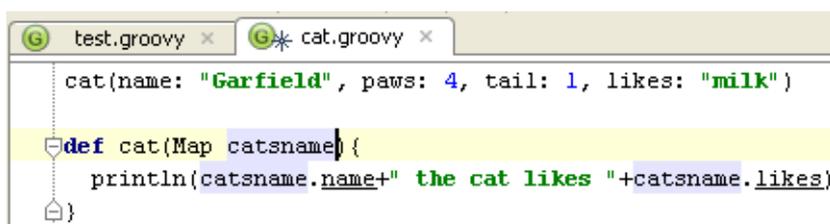
1. Place the caret at a parameter you want to convert, and press `Alt+Enter`:

2. On the context menu, choose **Convert parameter to map entry**, and specify conversion parameters:
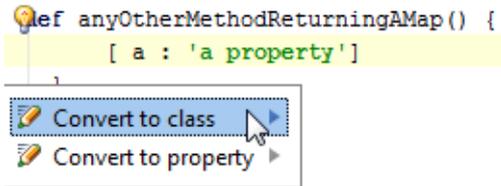

3. Repeat the procedure until all the required named arguments are represented as map entries. Note that all the subsequent conversions are done silently.

To convert Groovy map to a class instance

1. In a Groovy method, place the caret at a map to be converted, and press `Alt+Enter`:

2. On the context menu, choose **Convert to class**, and specify new class name and the package where the new class will be created. If such package doesn't yet exist, you can create one immediately,by clicking .

3. If necessary, choose to change return type of the method.

4. Click **OK** to apply changes and perform conversion.

   As a result, a class is created, with the fields corresponding to the keys of the original map.

To produce a list or a map

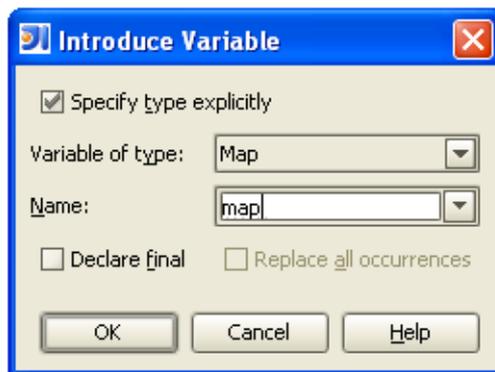1. Type contents of a list or a map in the editor:

```
[55, 127,-9,-100,568]
```

or

```
[fname:"John",lname:"Smith",age:31]
```

2. Place the caret somewhere inside the square brackets, and press `Ctrl+Alt+V`. The expression in brackets is converted to a list or map respectively.

To inline a list or a map

1. Place the caret on the list or map declaration, and press `Ctrl+Alt+N`. IntelliJ IDEA highlights the encountered usages:

```
Map map = [fname: "foo", lname: "bar", zzz: 31]

println map
```

2. Confirm inlining:

```
println() [fname: "foo", lname: "bar", zzz: 31]
```

See Also

Procedures:

- Instantiating Java Interface in Groovy

External Links:

- Quick prototyping in Groovy with Convert Map to Class intention ⧉

Web Resources:

- Developer Community ⧉