

Ever wanted to create a simple dropdown of all the Entity classes you have mapped in Doctrine? I have! For a recent project I needed to create a sample API form and load the entity type first, so I needed a way to properly load every Entity that's mapped programmatically and not with an ugly manually created array:

```
use Doctrine\Common\Inflector\Inflector;
```

```
private function getEntityChoices()
```

```
{
```

```
    $choices = array();
```

```
    $em = $this->getDoctrine()->getManager();
```

```
    $meta = $em->getMetadataFactory()->getAllMetadata();
```

```
    foreach ($meta as $m) {
```

```
        $r = new \ReflectionClass($m->getName());
```

```
        $name = $r->getShortName();
```

```
        $name = Inflector::tableize($name);
```

```
        $key = Inflector::pluralize($name);
```

```
        $choices[$key] = $r->getShortName();
```

```
    }
```

```
    ksort($choices);
```

```
    return $choices;
```

```
}
```

For this, I also applied Doctrine's Inflector so I could make the key value of my array the pluralized version of the entity name. This was useful for my purposes, but probably isn't a standard use case. Most use cases I imagine would simply want the simple class name:

```
$choices[strtolower($r->getShortName())] = $r->getShortName();
```

Share this:

- [LinkedIn](#)
- [Twitter](#)
- [Email](#)
- [Print](#)