

Important: Beta Release coming soon!

Source code for api.models.course



University of
Nebraska Tutoring
Portal

CSLC PORTAL BACKEND

[About](#)

[API Endpoints](#)

[Database Models](#)

[API Scripts](#)

[API Config](#)

CSLC PORTAL FRONTEND

[About](#)

```
from django.db import models
```

```
class Course(models.Model):
```

```
    """The course table holds all information about a specific course.
```

```
    Important:
```

```
        A course can have many sections, while a section can only be attributed to a
        The course represents what you would see in a catalog (e.g., CIST-1400 & CSC
        and sections are specific instances of the course (online, in person, etc.).
        Unlike sections, a course does not have a directly assigned professor;
        this data is stored in sections.
```

```
    Attributes:
```

```
        course_department (:obj:`CharField`): The department code of the course (e.g.
        course_name (:obj:`CharField`): The name of the course as it appears
        in the catalog (e.g., "Operating Systems").
        course_id (:obj:`IntegerField`): The unique identifier for the course.
        course_code (:obj:`CharField`): The code associated with the course (e.g., "
```

```
    Manager:
```

```
        generic (:obj:`Manager`): The default manager for the `Course` model.
```

```
    Example:
```

```
        Examples can be given using either the ``Example`` or ``Examples``
        sections. Sections support any reStructuredText formatting, including
        literal blocks::
```

```
        $ python example_google.py
```

```
    Todo:
```

```
        * For module TODOs
        * You have to also use ``sphinx.ext.todo`` extension
```

```
    """
```

```
    course_department = models.CharField(max_length=10, blank=False, default="CSCI")
    course_name = models.CharField(max_length=100, blank=False)
    course_id = models.IntegerField(blank=False, null=False)
    course_code = models.CharField(max_length=15, blank=False, null=False, unique=True)
```

```
    generic: models.Manager = models.Manager()
```

```
    def __str__(self) -> str:
```

```
        """(:obj:`str`): Returns the string representation of the course.
```

```
        Returns:
```

```
            str: _description_
```

```
    """  
    return str(self.course_name)
```

Made with [Sphinx](#) and @pradyunsg's [Furo](#)