

GraphQL Directive

[graphql](#), [directive](#), [deprecated](#), [skip](#), [include](#), [uppercase](#)

deprecated

code

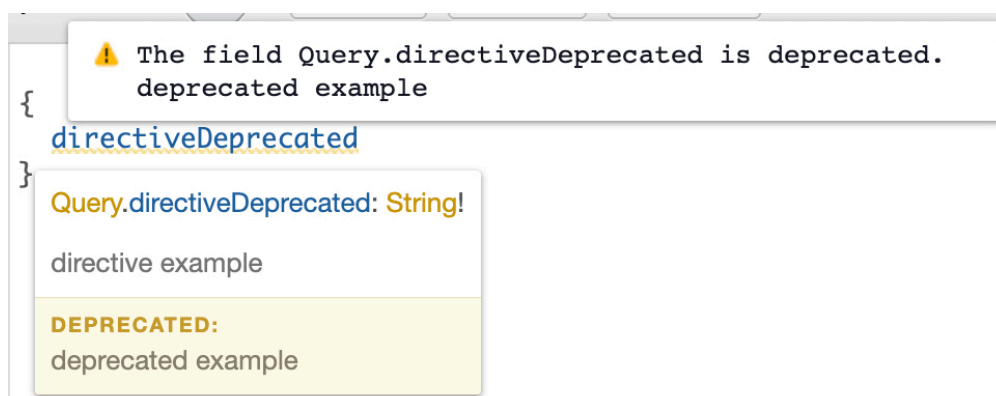
```
@strawberry.type
class DirectiveQuery:
    @strawberry.field(
        description="directive example", deprecation_reason="deprecated use
other field"
    )
    def directive_deprecated(self, info) -> str:
        return "result of directive_deprecated"
```

query

```
{
  directiveDeprecated
}
```

result

```
{
  "data": {
    "directiveDeprecated": "result of directive_deprecated"
  }
}
```



skip & include

code

```
@strawberry.type
class DirectiveQuery:
    @strawberry.field(description="directive skip")
    def directive_skip(self) -> str:
        return "skip"

    @strawberry.field(description="directive include")
    def directive_include(self) -> str:
        return "include"
```

query

```
query ($skip: Boolean!, $include: Boolean!) {
  directiveSkip @skip(if: $skip)
  directiveInclude @include(if: $include)
}
```

query variables

```
{"skip": false, "include": true}
```

result

```
{
  "data": {
    "directiveSkip": "skip",
    "directiveInclude": "include"
  }
}
```

uppercase

code

```
@strawberry.type
class DirectiveQuery:
    @strawberry.field(description="directive uppercase")
    def directive_uppercase(self) -> str:
        return "make uppercase"

@strawberry.directive(
    locations=[DirectiveLocation.FIELD], description="Make string uppercase"
)
def uppercase(value: str):
    return value.upper()

schema = strawberry.Schema(
    query=Query,
    mutation=Mutation,
    directives=[uppercase],
)
```

query

```
{
  directiveUppercase @uppercase
}
```

result

```
{
  "data": {
    "directiveUppercase": "MAKE UPPERCASE"
  }
}
```

query

```
{
  normal: directiveUppercase
  uppercase: directiveUppercase @uppercase
}
```

result

```
{
  "data": {
    "normal": "make uppercase",
    "uppercase": "MAKE UPPERCASE"
  }
}
```

replace

code

```
@strawberry.type
class Person:
    name: str = "Jess"
```

```
@strawberry.type
class DirectiveQuery:
    @strawberry.field
    def person(self) -> Person:
        return Person()
```

```
@strawberry.directive(locations=[DirectiveLocation.FIELD])
def replace(value: str, old: str, new: str):
    value = value.replace(old, new)
    return value
```

```
schema = strawberry.Schema(
    query=Query,
    mutation=Mutation,
    directives=[uppercase, replace],
)
```

query

```
{
  person {
    name @replace(old: "J", new: "John")
  }
}
```

```
}  
}
```

result

```
{  
  "data": {  
    "person": {  
      "name": "Johnness"  
    }  
  }  
}
```

ref

- https://github.com/strawberry-graphql/strawberry/blob/main/tests/schema/test_directives.py

Plugin Backlinks:

From:
<https://jace.link/> - **Various Ways**

Permanent link:
<https://jace.link/open/graphql-directive>

Last update: **2021/10/05 02:48**

