

```
from pydantic import BaseModel, Field
```

```
from typing import List
```

```
class FunctionSchema(BaseModel):
```

```
    name: str = Field(
```

```
        ...,
```

```
        title="Name",
```

```
        description="The name of the function.",
```

```
    )
```

```
    description: str = Field(
```

```
        ...,
```

```
        title="Description",
```

```
        description="The description of the function.",
```

```
    )
```

```
    parameters: BaseModel = Field(
```

```
        ...,
```

```
        title="Parameters",
```

```
        description="The parameters of the function.",
```

```
    )
```

```
class OpenAIFunctionCallSchema(BaseModel):
```

```
    """
```

```
    Represents the schema for an OpenAI function call.
```

Attributes:

type (str): The type of the function.

function (List[FunctionSchema]): The function to call.

"""

```
type: str = Field(
    "function",
    title="Type",
    description="The type of the function.",
)
function: List[FunctionSchema] = Field(
    ...,
    title="Function",
    description="The function to call.",
)
```