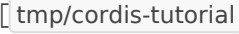
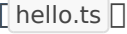


1.

[English](#) | 

 loader  apply  cordis  apply  ctx 



 tmp/cordis-tutorial  hello.ts 

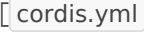
```
import type { Context } from '@deepseek-ai/cordis'

export const name = 'hello'

export function apply(ctx: Context) {
  console.log('hello from my first plugin')
}
```

name 



 cordis.yml 

- name: './hello.ts'

 C name  inject  loader 



node --import tsx ../../vendor/cordis/bin.js



hello from my first plugin



1. Context **Loader**
2. Loader [cordis.yml | ./hello.ts]
3. Cordis [apply(ctx)]

cordis.yml [dsh base] overlay []



☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ Cordis ☐ ☐ ☐ ☐ ☐ ☐

```
import { Service, type Context } from '@deepseek-ai/cordis'

// 1. Function plugin (what you just wrote).
export function apply(ctx: Context) {}

// 2. Object plugin: an object with an `apply` method.
export const objectPlugin = {
  name: 'object-plugin',
  apply(ctx: Context) {},
}

// 3. Class plugin: a Service subclass (covered in chapter 3).
export class MyService extends Service {
  constructor(ctx: Context) {
    super(ctx, 'myTutorialService')
  }
}
```

```
// 1. Function plugin (what you just wrote).
```

```
export function apply(ctx: Context) {}
```

```
// 2. Object plugin: an object with an `apply` method.
```

```
export const objectPlugin = {
```

```
name: 'object-plugin',
```

```
apply(ctx: Context) {},
```

}

```
// 3. Class plugin: a Service subclass (covered in chapter 3).
```

```
export class MyService extends Service {
```

```
constructor(ctx: Context) {
```

```
super(ctx, 'myTutorialService')
```

}

}



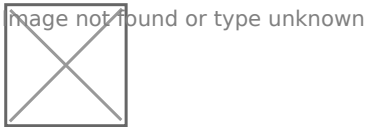
[apply]

```
export function apply(ctx: Context) {
  throw new Error('apply exploded')
}
```



 Cordis  logger  console 

 effect 



 #1
 3  2026 16:57:14
 3  2026 17:02:14