

# What is an Extension?

## Boston Dynamics explanation of an Extension

Extensions are software packages or static files introduced in 3.2 that can be installed or simply uploaded to the CORE I/O or Scout platforms.

Configured as software packages, Extensions provide the functionality for external developers to easily install applications onto CORE I/O and Scout platforms. On the CORE I/O, these software packages can integrate a newly-mounted payload with the Spot API, or not be associated with any payload functionality at all, such as uploading data to an AWS bucket.

Extensions can also simply be static files that developers need to upload into the CORE I/O or Scout platform. This configuration supports two important use cases:

1. It allows the developers to split their Extensions into smaller Extensions with the software components that need to be updated frequently and one or more larger Extensions with static files needed by the Extension with the software component. This configuration simplifies the process of updating Extensions by decoupling static large files from the frequently updatable files and installing them once, or less frequently.
2. It allows the developers to split their Extensions into a generic software package that is identical for all customers, and separate Extensions with configuration files that are applicable to one of a subset of customers. This configuration simplifies the process of installing customer-specific Extensions by maintaining the common part of the package in one Extension and the customer-specific configuration in another Extension.

## Our Explanation of an Extension

Extensions are a packaged group of software that we can use to run different docker images on SPOT. Each Extension can contain one or more docker images that are used to interface with hardware or perform computational tasks. This package format allows you to easily develop software on external machines and upload your complete package to the CORE I/O.

This setup is very similar to Docker's compose method. In fact, they even use docker-compose for running the services.

Extension files end in a .spx format and contains the following files:

- manifest.json
- docker-compose.yml
- [extension\_name].spx

- (optional) udev.rules
- (optional) Additional device files
- (optional) icon.png

## manifest.json

Contains the description and list of files to include in an extension.

Example:

```
{
  "description": "SENSEable Design Lab test.",
  "version": "0.1",
  "icon": "icon.png",
  "udev_rules": "senseable_udev.rules",
  "images": ["example.tgz"]
}
```

## docker-compose.yml

Contains the docker-compose files that builds and configures the docker files included in the extension.

Example:

```
services:
  ping_test:
    image: ping-test-dockerization-arm64
    ports:
      - "21002:21002"
```

## udev.rules

## additional device files

## icon.png

---

Revision #1

Created 17 October 2023 19:42:07 by Dwight Howard, II

Updated 17 October 2023 20:37:10 by Dwight Howard, II