

Aclaraciones sobre " Configuración de Karma y Jasmine para su correcto funcionamiento en los entornos de IC "

En este artículo se detallan los pasos necesarios para configurar **Karma** y **Jasmine** de manera adecuada, garantizando su correcto funcionamiento dentro de los entornos de **IC**. Esta configuración es esencial para asegurar que nuestras pruebas unitarias se ejecuten de forma efectiva y eficiente.

Subdirección de las Tecnologías de la Información y Comunicaciones

Área de Gobernanza y Calidad

Dirección General de Sistemas de Información y Comunicaciones

Áreas de Gobierno Tecnológico de SSII y del Dato



Servicio Andaluz de Salud
Consejería de Sanidad, Presidencia
y Emergencias

Tabla de Contenido

- Dirección General de Sistemas de Información y Comunicaciones
 - Áreas de Gobierno Tecnológico de SSII y del Dato
 - 1. Introducción
 - 1.1 Karma JS

Histórico de cambios

Los cambios en el documento vendrán acompañados de un registro de las modificaciones. De este modo se podrá realizar un seguimiento y consultar su evolución. Ordenándose de mas recientes a menos recientes, prestando especial cuidado a las cabeceras de la tablas dónde se indican las fechas de entrada en vigor y versión.

Versión	Pre-release	Adopción	Activa	Retiro	Alcance
v01r00	29 jun 2022				Versión inicial del documento

1. Introducción

Esta guía técnica aporta información sobre cómo configurar diferentes test runner en soluciones Front-End.

1.1 Karma JS

Karma, junto con Jasmine, permite probar el mismo código Javascript en varios navegadores web de una manera completamente automatizada.

Para su configuración se requiere establecer unos flags que dependerá de las necesidades concretas de cada proyecto Angular.

A continuación se define los flags más utilizados que el proveedor deberá probar y comprobar que funciona para la configuración de su proyecto en particular:

Flags	Descripción
--no-sandbox	Requerido para ejecutarse sin privilegios en docker. Cuando se ejecutan pruebas JS (usando Karma) con Headless Chrome en un contenedor docker, las pruebas fallan, informándose del siguiente error "Running as root without --no-sandbox is not supported".
--headless	habilita el modo headless de Chromes, sin interfaz gráfica.
--disable-dev-shm-usage	Solo se agrega cuando se establece la variable de entorno del sistema CI o cuando se encuentra dentro de una instancia de Docker. La partición /dev/shm es demasiado pequeña en ciertos entornos de VM, lo que hace que Chrome falle o se bloquee
--disable-web-security	Deshabilitar la seguridad web en el navegador.
--disable-software-rasterizer	Deshabilita el uso de Software 3D de rasterización (una serie de píxeles, puntos o líneas, que, cuando se muestran juntos, crean la imagen que se representó a través de formas)

--disable-gpu	Deshabilita la aceleración de hardware GPU en el servidor
----------------------	---

A continuación se muestra un ejemplo de configuración de Karma para desarrollo (modo Debug establecido a true)

1 Ejemplo configuración Karma: karma.conf.js

```
// Karma configuration file, see link for more information
// https://karma-runner.github.io/1.0/config/configuration-file.html
process.env.CHROME_BIN = require("puppeteer").executablePath();

module.exports = function (config) {
  config.set({
    basePath: "",
    frameworks: ["jasmine", "@angular-devkit/build-angular"],
    plugins: [
      require("karma-jasmine"),
      require("karma-chrome-launcher"),
      require("karma-jasmine-html-reporter"),
      require("karma-coverage"),
      require("@angular-devkit/build-angular/plugins/karma"),
      require("karma-spec-reporter"),
    ],
    client: {
      jasmine: {
        // you can add configuration options for Jasmine here
        // the possible options are listed at https://jasmine.github.io/api/edge/Configuration.html
        // for example, you can disable the random execution with `random: false`
        // or set a specific seed with `seed: 4321`
      },
      clearContext: false, // leave Jasmine Spec Runner output visible in browser
    },
    jasmineHtmlReporter: {
      suppressAll: true, // removes the duplicated traces
      suppressFailed: true, // Suppress failed messages
    },
    coverageReporter: {
      dir: require("path").join(__dirname, "./coverage"),
      subdir: ".",
      reporters: [{ type: "html" }, { type: "text-summary" }, { type: "lcov" }],
      check: {
        global: {
          statements: 65,
          branches: 65,
          functions: 65,
          lines: 65,
        },
      },
    },
    reporters: ["progress", "spec", "kjhtml"],
    port: 9876,
    colors: true,
    logLevel: config.LOG_INFO,
    autoWatch: true,
    browsers: ["ChromeHeadlessNoSandbox"],
    customLaunchers: {
      ChromeHeadlessNoSandbox: {
        base: "ChromeHeadless",
        flags: [
          "--no-sandbox", // required to run without privileges in docker
          "--user-data-dir=/tmp/chrome-test-profile",
          "--disable-web-security",
          "--remote-debugging-port=9222",
        ],
        debug: true,
      },
    },
    singleRun: false,
    restartOnFileChange: true,
  });
};
```

La configuración de Karma en Producción debe de contener únicamente los flags estrictamente necesarios. Por ejemplo, no debería de configurar los parámetros `--watch` o `--debug`

También es necesario que el proyecto esté configurado con las últimas versiones estables de Jasmine y Karma para que no se produzca errores como el siguiente por versiones deprecadas:

2 Ejemplo de error ejecución de test por versión deprecada de componentes

```
> ng test
```

```
'karma-coverage-istanbul-reporter' usage has been deprecated since version 11.
```