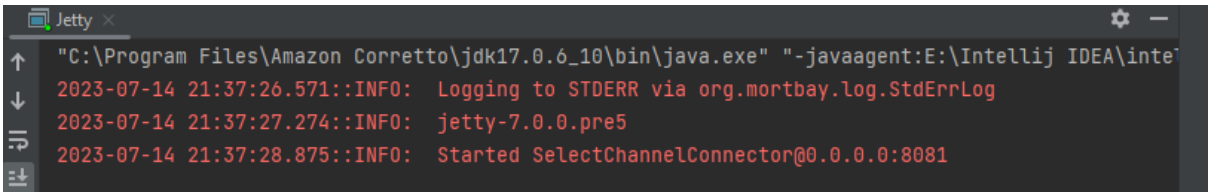
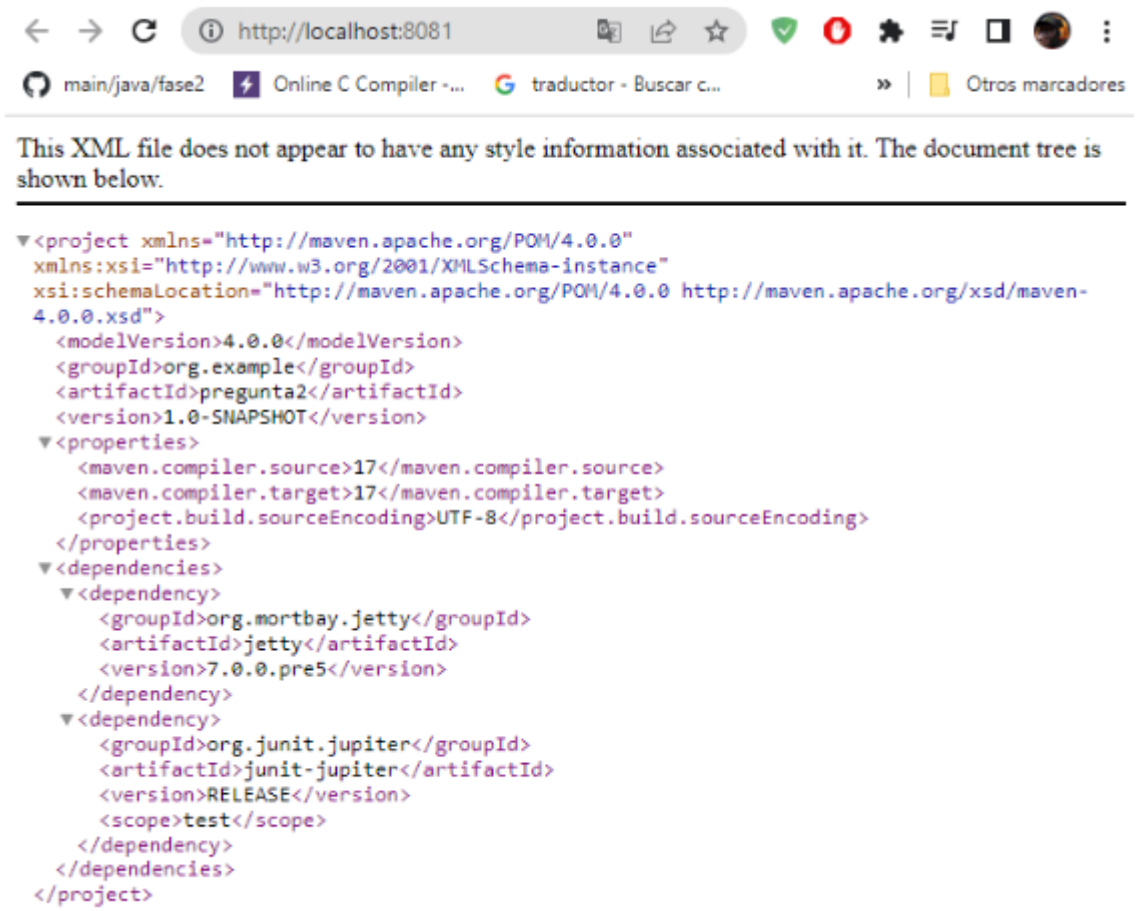
## PREGUNTA 2:

**PREGUNTA:**
Jetty es un servidor HTTP basado en Java y un contenedor de Servlets, para poder ejecutarlo primero agrego algunas dependencias en mi maven. Le damos en ejecutar:




**PREGUNTA:**

- **Explica los resultados de ejecutar este código.**

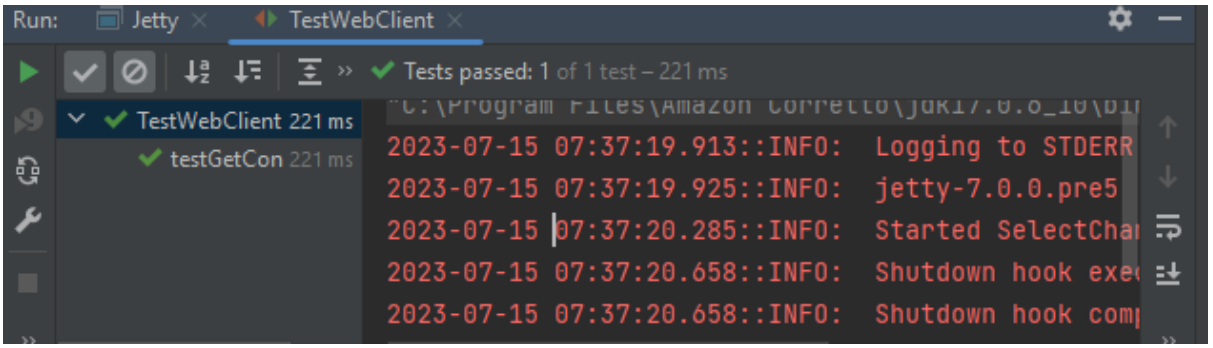
Agregamos en nuestra clase Jetty un inner class TestGetContentOkHandler que heredara de la interfaz AbstractHandler de Jetty. Importaremos algunas Bibliotecas para esa claseInterna ( usaremos las clases HttpServletRequest, HttpServletResponse, HttpHeaders, OutputStream)
Todo esto para manejar una solicitud HTTP y generar una respuesta exitosa.

**PREGUNTA:**

- Ahora que este controlador está escrito, podemos decirle a Jetty que lo use llamando a context.setHandler(new TestGetContentOkHandler()). Escribe una prueba llamada **Testwebclient.java** y explica los resultados.

Creamos un test “TestWebInicial”, el se encarga de probar el método getContent() de nuestra clase WebClient. El propósito de este test es verificar que el cliente pueda obtener correctamente el contenido de una URL específica.

El resultado de este test depende de si el método getContent() de WebClient puede obtener el contenido correctamente, en nuestro caso si.
Esta imagen de la salida nos indica que el servidor Jetty se inició con éxito, el conector se estableció en el puerto 8081 y el servidor se apagó después de que las pruebas finalizaron.




**PREGUNTA:**

- Analiza el uso de stubs en conexión HTTP. Se quiere probar el código de forma aislada. Las pruebas funcionales o de integración probarán la conexión en una etapa posterior.

Cree una clase StubHttpURLConnection, esto nos permitira simular diferentes respuestas y escenarios para evaluar el comportamiento del código bajo prueba sin la necesidad de realizar conexiones reales a través de HTTP, se puede lograr redirigiendo las llamadas a la clase HttpURLConnection a nuestra propia clase de STUB.

```java
import java.io.ByteArrayInputStream;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.URL;

public class StubHttpURLConnection extends HttpURLConnection {
    private boolean connected;

    public StubHttpURLConnection(URL url) {
        super(url);
        connected = false;
    }

    @Override
    public void disconnect() {
    }

    @Override
    public boolean usingProxy() {
        return false;
    }

    @Override
    public void connect() {
        if (!connected) {
            connected = true;
        }
    }

    @Override
    public InputStream getInputStream() {
        String response = "Esto funciona";
        return new ByteArrayInputStream(response.getBytes());
    }
}
```

**PREGUNTA:**

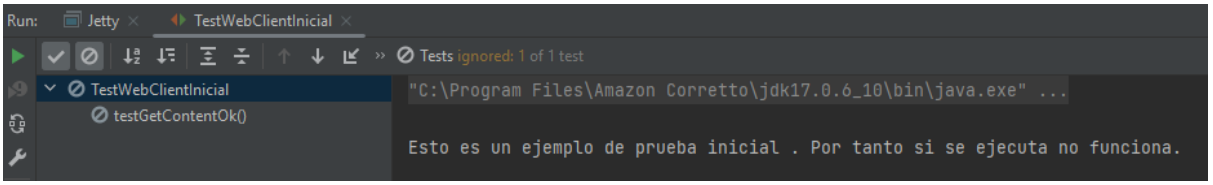
- Para implementar un controlador de protocolo de URL personalizado, se llama al método estático de URL **setURLStreamHandlerFactory** y se le pasa un **URLStreamHandlerFactory** personalizado. Realiza la implementación de código auxiliar del controlador de flujo de URL de manera que cada vez que se llama al método URL openConnection, se llama a la clase URLStreamHandlerFactory para devolver un URLStream-Handler.

Se creo la clase **“StubStreamHandlerFactory”**,

```java
import java.io.IOException;
import java.net.URL;
import java.net.URLConnection;
import java.net.URLStreamHandler;
import java.net.URLStreamHandlerFactory;

public class StubURLStreamHandlerFactory implements URLStreamHandlerFactory {

    @Override
    public URLStreamHandler createURLStreamHandler(String protocol) {
        if ("http".equals(protocol)) {
            return new StubHttpURLStreamHandler();
        }
        return null;
    }

    private static class StubHttpURLStreamHandler extends URLStreamHandler {
        @Override
        protected URLConnection openConnection(URL url) throws IOException {
            return new StubHttpURLConnection(url);
        }
    }
}
```

**PREGUNTA**:

- Realiza una implementación stub de la clase HttpURLConnection para que podamos devolver cualquier valor que queramos para la prueba. Esta implementación simple devuelve la cadena "Esto funciona" . **¿Se pasa la prueba?.**




En clase WebClient , agregue un metodo setConnection , asi para cuando hacemos el test , le mande el STUB.
Si pasa la prueba.