Preguntas

¿Cuál es la diferencia entre el propósito y el contenido de Gemfile y Gemfile.lock? ¿Qué archivo se necesita para reproducir completamente las gemas del entorno de desarrollo en el entorno de producción?

El Gemfile se utiliza para especificar las gemas y sus versiones en tu proyecto mientras que el Gemfile.lock registra las versiones exactas de las gemas que se instalaron en desarrollo. Para reproducir el entorno de desarrollo en producción, solo necesitas copiar y utilizar el Gemfile.lock en tu entorno de producción.

¿Qué sucede si intentas visitar una URL no raíz cómo https://localhost:3000/hello y por qué? (la raíz de tu URL variará)
No coincidirá con ninguna de las rutas definidas en tu aplicación y, por lo tanto, provocará un error que mandara el siguiente error “Sinatra doesn’t know this ditty.” esto se debe a que no se ha definido una ruta específica para "/hello" en tu aplicación.
Además, en tu terminal donde pruebas la aplicación aparecerá un mensaje similar al siguiente:

```
ERROR bad Request-Line
`\x16\x03\x01\x02\x00\x01\x00\x01�\x03\x03�?�\x10z�ï\x04�aב�bSIy��A�/\a5�\bP�a\
x02� ]�x\x0E�ᵽ�w␣��労?a\x15�\x10vN����}��(�\x00
jj\x13\x01\x13\x02\x13\x03�+�/�,�Q̧�\x13�\x14\x00�\x00�\x00/\x005\x01\x00\x01���\x
00\x00\x00\r\x00\x14\x00\x12\x04\x03\b\x04\x04\x01\x05\x03\b\x05\x05\x01\b\x06\x06\x0
1\x02\x01\x00-\x00\x02\x01\x01\x00\x10\x00\x0E\x00\f\x02h2\bhttp/1.1\x00#\x00\x00\x0
0\x12\x00\x00\x00\x17\x00\x00�\x01\x00\x01\x00Di\x00\x05\x00\x03\x02h2\x00\x00\x0
0\x0E\x00\f\x00\x00\tlocalhost\x00'.
```

Lo cual parece ser una respuesta de error HTTP, señalando que el navegador ha recibido una respuesta inesperada del servidor en "localhost:3000".

¿Cómo se le dice a un entorno de producción cómo iniciar un servidor de aplicaciones u otros procesos necesarios para recibir solicitudes e iniciar su aplicación?

Preguntas

Según los casos de prueba, ¿cuántos argumentos espera el constructor de la clase de juegos (identifica la clase) y, por lo tanto, cómo será la primera línea de la definición del método que debes agregar a wordguesser_game.rb?
Un argumento (en este ejemplo, "glorp"), y dado que los constructores en Ruby siempre se denominan initialize, la primera línea será def initialize(new_word)o algo similar.

Según las pruebas de este bloque describe, ¿qué variables de instancia se espera que tenga WordGuesserGame?
@word, @guesses, y @wrong_guesses.

Pregunta

Enumera el estado mínimo del juego que se debe mantener durante una partida de Wordguesser.

La palabra secreta; la lista de letras que se han adivinado correctamente; la lista de letras que han sido adivinadas incorrectamente. Convenientemente, la clase WordGuesserGame, bien factorizada, encapsula este estado utilizando sus variables de instancia, como recomienda un diseño orientado a objetos adecuado.

Pregunta

Enumera las acciones del jugador que podrían provocar cambios en el estado del juego.
Adivina una letra: posiblemente modifica las listas de aciertos o errores; posiblemente resulte en ganar o perder el juego.

Iniciar un nuevo juego: elige una nueva palabra y deja vacías las listas de acertijos incorrectos y correctos.

Pregunta

Para un buen diseño RESTful, ¿cuáles de las operaciones de recursos deberían ser manejadas por HTTP GET y cuáles deberían ser manejadas por HTTP POST?

Las operaciones manejadas con GETno deberían tener efectos secundarios en el recurso, por lo que showpueden ser manejadas por a GET, pero createy guess(que modifican el estado del juego) deben usarse POST. (De hecho, en una verdadera arquitectura orientada a servicios también podemos optar por utilizar otros verbos HTTP como PUTy DELETE, pero no cubriremos eso en esta tarea).

Preguntas

¿Por qué es apropiado que la nueva acción utilice GET en lugar de POST?

La newacción por sí sola no provoca ningún cambio de estado: simplemente devuelve un formulario que el jugador puede enviar.

Explica por qué la acción GET /new no sería necesaria si tu juego Wordguesser fuera llamado como un servicio en una verdadera arquitectura orientada a servicios.
En una verdadera SOA, el servicio que llama a Wordguesser puede generar una POSTsolicitud HTTP directamente. El único motivo de la newacción es proporcionar al usuario web humano una forma de generar esa solicitud.

Pregunta

@game en este contexto es una variable de instancia de qué clase?

Es una variable de instancia de la WordGuesserAppclase en el archivo app.rb. Recuerde que aquí estamos tratando con dos clases de Ruby: la WordGuesserGameclase encapsula la lógica del juego en sí (es decir, el modelo en modelo-vista-controlador), mientras que WordguesserAppencapsula la lógica que nos permite entregar el juego como SaaS (puede pensar en ello a grandes rasgos). como la lógica del controlador más la capacidad de representar las vistas mediante erb).

Pregunta

¿Por qué esto ahorra trabajo en comparación con simplemente almacenar esos mensajes en el hash de sesion []?
Cuando ponemos algo allí, session[]permanece allí hasta que lo eliminamos. El caso común de un mensaje que debe sobrevivir a una redirección es que sólo debe mostrarse una vez; flash[]Incluye la funcionalidad adicional de borrar los mensajes después de la siguiente solicitud.

Pregunta

Según el resultado de ejecutar este comando, ¿cuál es la URL completa que debes visitar para visitar la página New Game?

El código Ruby get '/new' do...muestra app.rbla página Nuevo juego, por lo que la URL completa tiene el formatohttp://localhost:3000/new

Pregunta

¿Dónde está el código HTML de esta página?

Está en views/new.erb, que la directiva procesa en HTML erb :new.

Preguntas

Lea la sección sobre " Using Capybara with Cucumber” en la página de inicio de Capybara. ¿Qué pasos utiliza Capybara para simular el servidor como lo haría un navegador? ¿Qué pasos utiliza Capybara para inspeccionar la respuesta de la aplicación al estímulo?
Las definiciones de pasos que utilizan visit, simulan un navegador visitando una página y/o completando un formulario en esa página y haciendo clic en sus botones click_button. fill_inLos que lo utilizan have_contentestán inspeccionando la salida.

Mirando features/guess.feature, ¿cuál es la función de las tres líneas que siguen al encabezado "Feature:"?
Son comentarios que muestran el propósito y los actores de esta historia. El pepino no los ejecutará.

En el mismo archivo, observando el paso del escenario Given I start a new game with word "garply" qué líneas en game_steps.rb se invocarán cuando Cucumber intente ejecutar este paso y cuál es el papel de la cadena "garply" en el paso?
Se ejecutarán las líneas 13-16 del archivo. Dado que un paso se elige haciendo coincidir una expresión regular, wordcoincidirá con el primer (y en este caso único) grupo de captura de paréntesis en la expresión regular, que en este ejemplo es garply.

Pregunta

Cuando el "simulador de navegador" en Capybara emite la solicitud de visit '/new', Capybara realizará un HTTP GET a la URL parcial /new en la aplicación. ¿Por qué crees que visit siempre realiza un GET, en lugar de dar la opción de realizar un GET o un POST en un paso determinado?
Se supone que Cucumber/Capybara solo puede hacer lo que un usuario humano puede hacer. Como comentamos anteriormente, la única forma en que un usuario humano puede hacer que se realice una POST a través de un navegador web es enviando un formulario HTML, lo cual se logra en click_buttonCapybara.

Pregunta

¿Cuál es el significado de usar Given versus When versus Them en el archivo de características? ¿Qué pasa si los cambias? Realiza un experimento sencillo para averiguarlo y luego confirme los resultados utilizando Google.
Todas las palabras clave son alias para el mismo método. Cuál utilice está determinado por lo que hace que el escenario sea más legible.

Pregunta

En game_steps.rb, mira el código del paso "I start a new game..." y, en particular, el comando stub_request. Dada la pista de que ese comando lo proporciona una gema (biblioteca) llamada webmock, ¿qué sucede con esa línea y por qué es necesaria? (Utiliza Google si es necesario).
Webmock permite que nuestras pruebas "intercepten" solicitudes HTTP provenientes **de** nuestra aplicación y dirigidas a otro servicio. En este caso, intercepta la solicitud POST (la misma que hiciste manualmente curlen una parte anterior de la tarea) y falsifica el valor de respuesta. Esto nos permite imponer un comportamiento determinista de nuestras pruebas y también significa que no accedemos al servidor externo real cada vez que se ejecuta nuestra prueba.

Pregunta

En tu código Sinatra para procesar una adivinación, ¿qué expresión usaría para extraer *solo el primer carácter* de lo que el usuario escribió en el campo de adivinación de letras del formulario en show.erb?
params[:guess].to_s[o]o su equivalente. to_sse convierte nila la cadena vacía en caso de que el campo del formulario se haya dejado en blanco (y por lo tanto no se incluya en paramsabsoluto). [o]toma solo el primer carácter; para una cadena vacía, devuelve una cadena vacía.