

Universidad de San Carlos de Guatemala

Facultad de ingeniería

Escuela de Ciencias y Sistemas

Organización de lenguajes y compiladores 2

Segundo semestre 2023

Manual Técnico

Jhonatan Josué Tzunun Yax

Introducción

En el siguiente manual se describe a detalle los procedimientos y funciones, así como el contenido de cada una y porque fue necesaria la creación de esta, para que se usó cada procedimiento, y función, explicando de forma breve la lógica aplicada en este proyecto.

Endpoints

- router.HandleFunc("/run", server.InputOutput)
- router.HandleFunc("/tree", server.TreeParse)

run

```
func InputOutput(w http.ResponseWriter, r *http.Request) {
    request := CodeEntry{}
    response := AllInOne{}
    // var aux []environment.Error_
    err := json.NewDecoder(r.Body).Decode(&request)
    if err != nil {
        w.WriteHeader(http.StatusBadRequest)
        fmt.Fprintf(w, "%v", err)
        return
    }
    response.Message, response.Error, response.Symbol = analyzer(request.Code)
    // fmt.Println(aux)
    w.Header().Set("content-type", "application/json")
    w.WriteHeader(http.StatusOK)
    json.NewEncoder(w).Encode(response)
}
```

Recibe el código de entrada y devuelve los print, la tabla de errores y la tabla de símbolos.

TreeParse

```

func TreeParse(w http.ResponseWriter) {
    url := "http://lab.antlr.org/pa
    /***** CODIGO DE ENTRADA
    request := CodeEntry{}
    err := json.NewDecoder(r.Body).Decode(&request)
    if err != nil {
        w.WriteHeader(http.StatusBadRequest)
        fmt.Fprintf(w, "%v", err)
        return
    }
    /* ***** */
    grammar := readFile("other/grammar.txt")
    lexgrammar := readFile("other/lexgrammar.txt")
    res := &Tree{}

    data := Data{
        Grammar:    grammar,
        Lexgrammar: lexgrammar,
        Input:       request.Code,
        Start:       "s",
    }
    content, err := json.Marshal(data)
    resp, err := http.Post(url, "Content-Type: application/json", bytes.NewBuffer(content))
    if err != nil {
        fmt.Println("Error al hacer la solicitud:", err)
        return
    }
    defer resp.Body.Close()

    err = json.NewDecoder(resp.Body).Decode(res)
    if err != nil {
        fmt.Println("Error al leer la respuesta:", err)
    }
}

```

Decode reads the next JSON-encoded value from its input and stores it in the value pointed to by v.

See the documentation for Unmarshal for details about the conversion of JSON into a Go value.

(json.Decoder).Decode on pkg.go.dev

You, 1 second ago · Uncommitted changes

Recibe el código de entrada y devuelve un json que contiene el svg del árbol de análisis concreto.