# Практика 11 HTTP, API, FastAPI

Ращупкин ЕВ КЭ-120

Реализуйте любое приложение с API

Подготовьте отчет со скриншотами выполненного подключения и фрагментами кода

1. Получение всех анекдотов

Метод: GET

Путь: /anecdotes

Хэндлер: getAnecdote

router.GET("/anecdotes", getAnecdote)

func getAnecdote(c \*gin.Context) {

c.IndentedJSON(http.StatusOK, anecdotes)

}

2. Получение одного анекдота по ID

Метод: GET

Путь: /anecdotes/:id

Хэндлер: getAnecdoteByID

router.GET("/anecdotes/:id", getAnecdoteByID)

func getAnecdoteByID(c \*gin.Context) {

id := c.Param("id")

for \_, a := range anecdotes {

if a.ID == id {

c.IndentedJSON(http.StatusOK, a)

return

}

}

c.IndentedJSON(http.StatusNotFound, gin.H{"message": "anecdote not found"})

}

3. Создание нового анекдота

Метод: POST

Путь: /anecdotes

Хэндлер: postAnecdote

router.POST("/anecdotes", postAnecdote)

func postAnecdote(c \*gin.Context) {

var newAnecdote anecdote

if err := c.BindJSON(&newAnecdote); err != nil {

return

}

anecdotes = append(anecdotes, newAnecdote)

c.IndentedJSON(http.StatusCreated, newAnecdote)

}

4. Полное обновление анекдота

Метод: PUT

Путь: /anecdotes/:id

Хэндлер: updateAnecdote

router.PUT("/anecdotes/:id", updateAnecdote)

func updateAnecdote(c \*gin.Context) {

id := c.Param("id")

var updatedAnecdote anecdote

if err := c.BindJSON(&updatedAnecdote); err != nil {

return

}

for i, a := range anecdotes {

if a.ID == id {

updatedAnecdote.ID = id

anecdotes[i] = updatedAnecdote

c.IndentedJSON(http.StatusOK, updatedAnecdote)

return

}

}

c.IndentedJSON(http.StatusNotFound, gin.H{"message": "anecdote not found"})

}

5. Частичное обновление анекдота

Метод: PATCH

Путь: /anecdotes/:id

Хэндлер: patchAnecdote

router.PATCH("/anecdotes/:id", patchAnecdote)

func patchAnecdote(c \*gin.Context) {

id := c.Param("id")

var updatedFields anecdote

if err := c.BindJSON(&updatedFields); err != nil {

return

}

for i, a := range anecdotes {

if a.ID == id {

// обновляем только непустые поля

if updatedFields.Title != "" { a.Title = updatedFields.Title }

if updatedFields.Author != "" { a.Author = updatedFields.Author }

if updatedFields.Text != "" { a.Text = updatedFields.Text }

if updatedFields.Likes != 0 { a.Likes = updatedFields.Likes }

if updatedFields.Dislikes != 0 { a.Dislikes = updatedFields.Dislikes }

anecdotes[i] = a

c.IndentedJSON(http.StatusOK, anecdotes[i])

return

}

}

c.IndentedJSON(http.StatusNotFound, gin.H{"message": "anecdote not found"})

}

6. Удаление анекдота

Метод: DELETE

Путь: /anecdotes/:id

Хэндлер: deleteAnecdoteByID

router.DELETE("/anecdotes/:id", deleteAnecdoteByID)

func deleteAnecdoteByID(c \*gin.Context) {

id := c.Param("id")

for i, a := range anecdotes {

if a.ID == id {

anecdotes = slices.Delete(anecdotes, i, i+1)

c.IndentedJSON(http.StatusOK, gin.H{"message": "anecdote deleted"})

return

}

}

c.IndentedJSON(http.StatusNotFound, gin.H{"message": "anecdote not found"})

}

7. Лайки Дизлайки

Метод: POST

Пути:

/anecdotes/:id/like → likeAnecdote

/anecdotes/:id/dislike → dislikeAnecdote

router.POST("/anecdotes/:id/like", likeAnecdote)

router.POST("/anecdotes/:id/dislike", dislikeAnecdote)

func likeAnecdote(c \*gin.Context) {

id := c.Param("id")

for i, a := range anecdotes {

if a.ID == id {

anecdotes[i].Likes++

c.IndentedJSON(http.StatusOK, anecdotes[i])

return

}

}

c.IndentedJSON(http.StatusNotFound, gin.H{"message": "anecdote not found"})

}

func dislikeAnecdote(c \*gin.Context) {

id := c.Param("id")

for i, a := range anecdotes {

if a.ID == id {

anecdotes[i].Dislikes++

c.IndentedJSON(http.StatusOK, anecdotes[i])

return

}

}

c.IndentedJSON(http.StatusNotFound, gin.H{"message": "anecdote not found"})

}

Запуск сервера

router.Run("localhost:8080")

Все маршруты подключены через движок Gin, а данные хранятся в срезе anecdotes в памяти.

Пример работы:

