



Pencil

Paul Hewitt & Ian Quach

10/10/19



What are we doing?

- Lightweight, flexible customer scheduling SaaS
- Highly modular
- Open to all industries that need/manage scheduling
- Have multiple, real customers that have expressed interest
- Serverless



How we stack up

	Pencil	YouCanBook	SquareSpace
Gamification/Rewards	✓	✗	✗
SMS Alerts	✓	✓	✗
Third Party Integrations	✓	✓	✗
Customer Tracking	✓	✗	✗

What have we been up to?



- Setting up the backend
- Configuring AWS
- Decided on a name (finally)
- UI/UX considerations
- First steps with project management



AWS

- S3
- Route 53 (Possibly)
- Dynamo
- Lambda
- API Gateway
- Cloudwatch Logs

```

package main
import (
    "context"
    "encoding/json"
    "fmt"
    "github.com/aws/aws-lambda-go/events"
    "github.com/aws/aws-lambda-go/lambda"
    "strings"
)
func Handler(_ context.Context, request events.APIGatewayProxyRequest)
(events.APIGatewayProxyResponse, error) {
    c, err := controller.NewController()
    if err != nil {
        return errorHandler.NewErrorResponse(err, 500), nil
    }
    var prop string
    for k := range request.Headers {
        switch strings.ToLower(k) {
            case "prop":
                prop = request.Headers[k]
        }
    }
    for k := range request.QueryStringParameters {
        switch strings.ToLower(k) {
            case "prop":

```

```

                prop = request.QueryStringParameters[k]
            }
        }
        response, err := c.Function()
        if err != nil {
            return errorHandler.NewErrorResponse(err, 404), nil
        }
        body, err := json.Marshal(models.RequestBody{
            Property: *response,
        })
        if err != nil {
            return errorHandler.NewErrorResponse(err, 500), nil
        }
        resp := events.APIGatewayProxyResponse{
            StatusCode: 200,
            IsBase64Encoded: false,
            Body: string(body),
            Headers: map[string]string{
                "Content-Type": "application/json",
            },
        }
        return resp, nil
    }
}
func main() {
    lambda.Start(Handler)
}

```



Reflection

- So far on track!
- No apparent barriers to success
- GitHub Tour