To create a Lambda function that simulates rolling a dice and can be called with HTTPS requests from anywhere, you would need to do the following:

1. Create a new Lambda function using AWS Lambda. You can do this through the AWS Management Console or by using the AWS CLI.
2. Write the code for the Lambda function using a programming language such as Python or Node.js. The function should take an input parameter "sides" (number of sides of the dice) and generate a random number within the range of 1 to "sides"
3. Attach an API Gateway trigger to the Lambda function, This will allow the function to be called with HTTPS requests. You can also add an API key for extra security
4. Deploy your API to a stage, you can test this endpoint by sending a GET request to the endpoint with the query parameter "sides" and the number of sides of the dice you want to roll
5. Add an IAM role to the Lambda function that allows the function to be invoked from anywhere, this allows the function to be called with HTTPS requests from anywhere.

Here is an example of a simple Python function that simulates rolling a dice with n-sides:

| import random  def lambda\_handler(event, context):  sides = int(event["queryStringParameters"]["sides"])  roll = random.randint(1, sides)  return {  "statusCode": 200,  "body": f"You rolled a {roll} on a {sides}-sided dice"  } |
| --- |

To attach an API Gateway trigger to the Lambda function, you can do the following:

1. Open the AWS Management Console, navigate to the Lambda service, and select the Lambda function you created.
2. Under the Designer section, click on the “Add trigger” button.
3. In the “Add trigger” screen, select “API Gateway” from the list of available triggers.
4. In the “Configure triggers” screen, select “Create a new API” and choose the “HTTP API” option. This will create a new API Gateway that will be connected to your Lambda function.
5. In the “Security” section, you can choose to add an API key if you want to add extra security.
6. Click on the “Add” button to save the trigger and link the API Gateway to your Lambda function.

Once the trigger is added, your Lambda function will be invoked when an incoming request is made to the API Gateway. You can test this by sending a GET request to the endpoint generated by the API Gateway.

Please note that the specific steps may vary depending on the version of the AWS Management Console you are using, but the overall process should be similar. Also, be aware that there are costs associated with using the API Gateway service.

| exports.handler = async (event) => {  let sides = event.queryStringParameters.sides;  let roll = Math.floor(Math.random() \* sides) + 1;  return {  statusCode: 200,  body: `You rolled a ${roll} on a ${sides}-sided dice`  } }; |
| --- |

An "Internal Server Error" typically indicates that there is an issue with the Lambda function itself, rather than with the API Gateway or the client making the request.

Here are a few things you can try to troubleshoot the issue:

* Check the CloudWatch logs for the Lambda function to see if there are any error messages or stack traces that may provide more information about the cause of the error.
* Make sure that the Node.js runtime version you are using is compatible with the version of Node.js you are using to develop the function.
* Check if the query parameter is being passed correctly and if it is being passed as a number or a string.
* Ensure that the handler name in the AWS Lambda Configuration page is set correctly to the function you are using.
* Make sure that the Lambda function has the appropriate permissions to execute and access any resources it needs.
* If you are using a VPC, make sure the function has the correct VPC settings and can access the internet.

If none of the above solutions helped, you can also check the API Gateway logs, this can give more information about the request and the error.