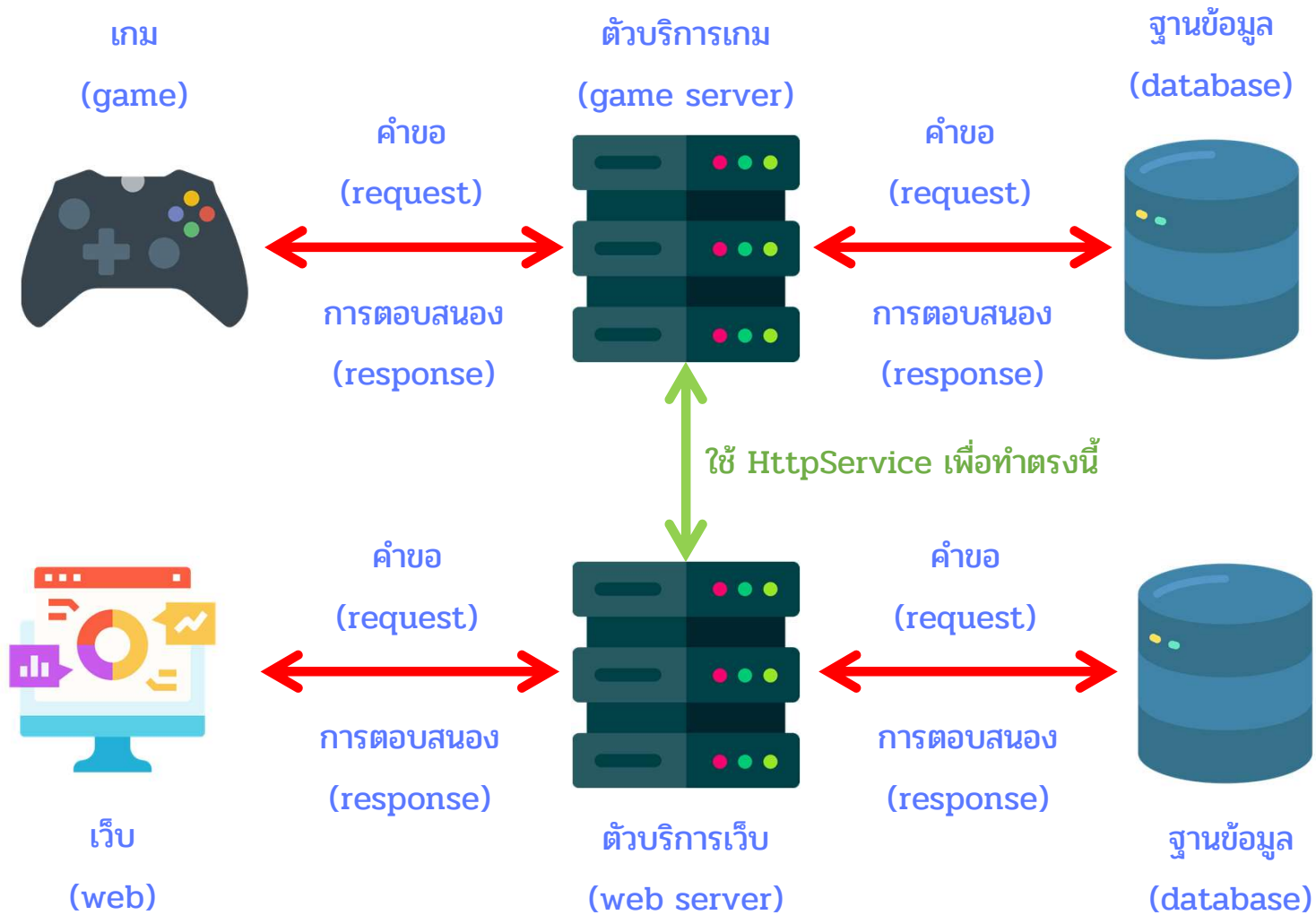


HttpService

HttpService คืออะไร

HttpService เป็น**บริการ (service)** หนึ่งของ Roblox Engine โดย HttpService จะยอมให้ส่งคำขอ HTTP จาก**ตัวบริการเกม (game server)** โดยใช้ **RequestAsync**, **GetAsync** และ **PostAsync** ซึ่งบริการนี้ช่วยให้เกมสามารถรวมเข้ากับบริการเว็บนอก Roblox ได้

หมายเหตุ ควรส่งคำขอ HTTP ไปยังแพลตฟอร์มบุคคลภายนอกที่เชื่อถือได้เท่านั้น เพื่อหลีกเลี่ยงความเสี่ยงด้านความปลอดภัยของตัวเกม



การเปิดใช้งานคำขอ HTTP

การเปิดใช้งานคำขอ HTTP

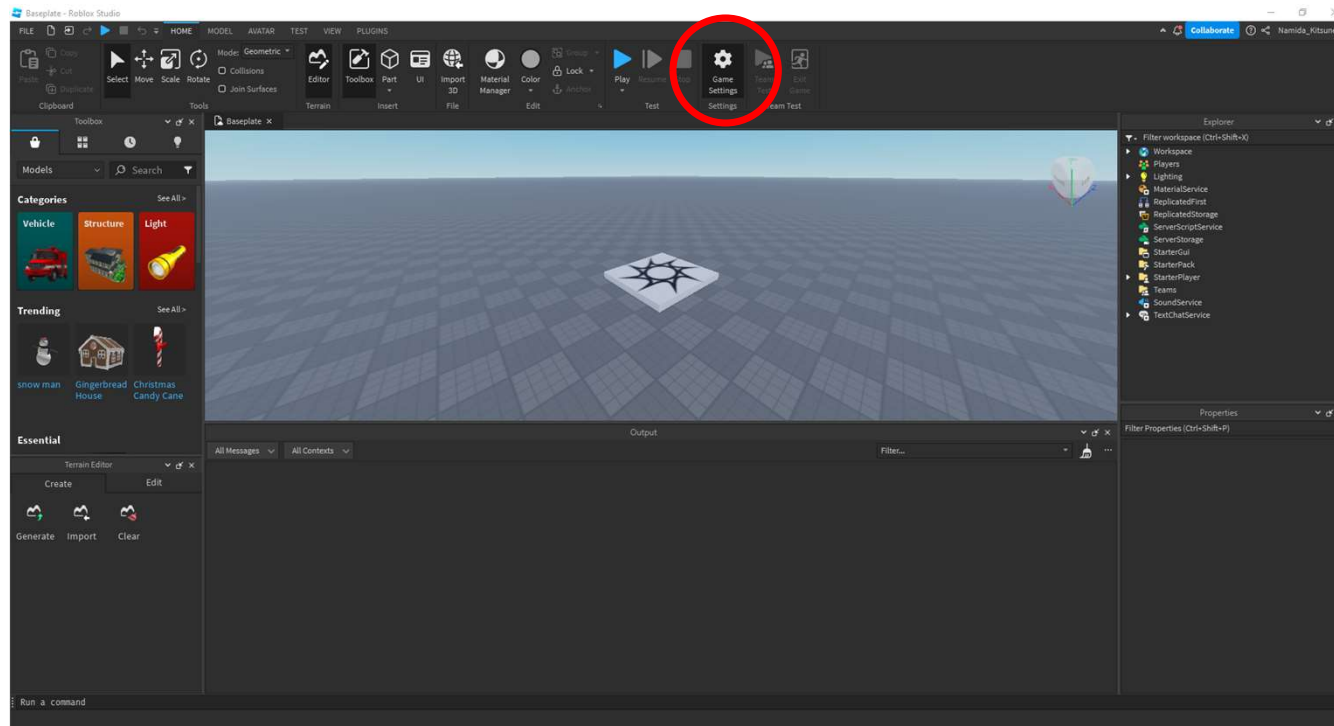
โดยปกติแล้วฟังก์ชันการส่งคำขอ HTTP ไม่ได้ถูกเปิดใช้งานเป็นค่าเริ่มต้น หากพยายามใช้งานการส่งคำขอ HTTP ในขณะที่ปิดการใช้งาน ผลลัพธ์ที่ได้จะเป็นข้อผิดพลาดว่า

“Http requests are not enabled. Enable via game settings”

วิธีบันทึกเกมไว้ใน Roblox
(ใครทำแล้ว สามารถข้ามเนื้อหานี้ได้)

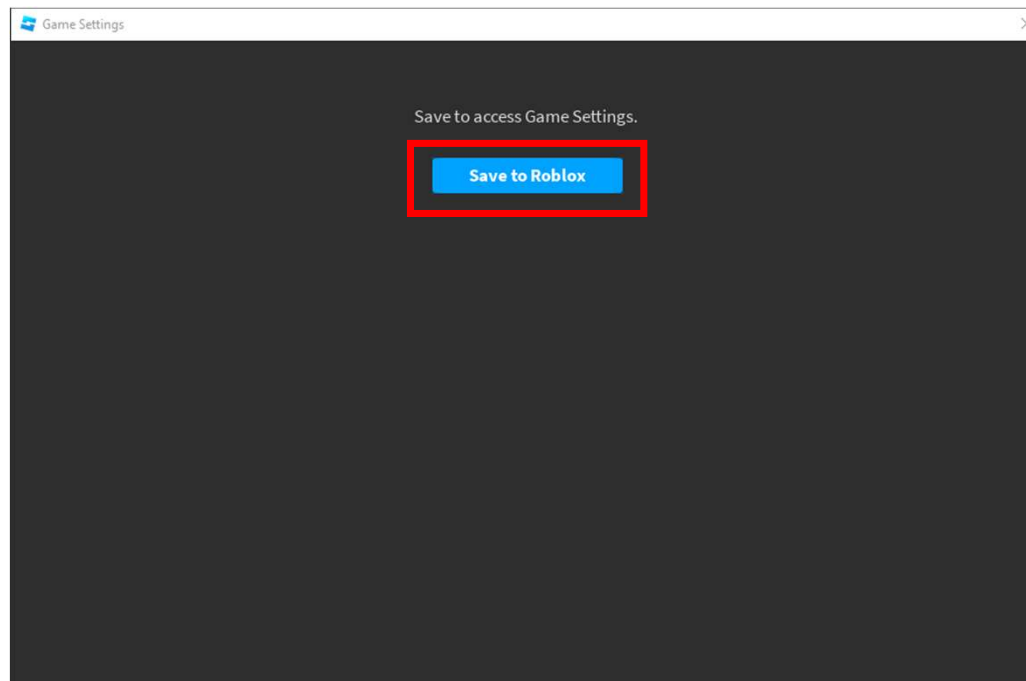
วิธีบันทึกเกมไว้ใน Roblox

คลิก Game Settings



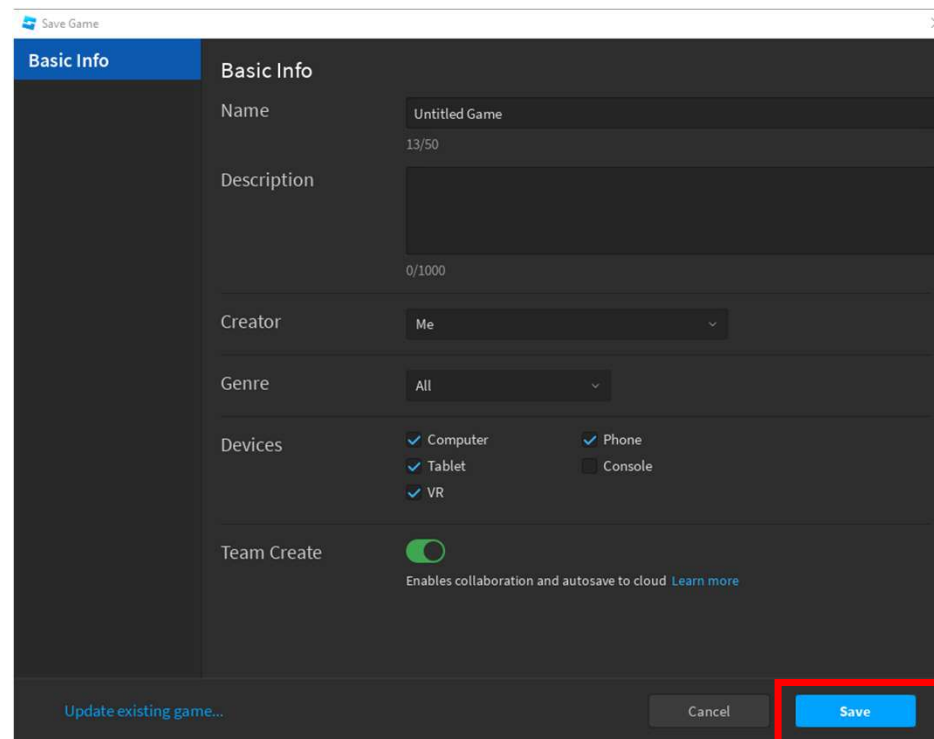
วิธีบันทึกเกมไว้ใน Roblox (ต่อ)

ถ้ามีหน้าต่างนี้ คลิก Save to Roblox เพื่อเข้าถึงการตั้งค่าเกม



วิธีบันทึกเกมไว้ใน Roblox (ต่อ)

ถ้ามีหน้าต่างนี้ตั้งค่าสิ่งต่าง ๆ ตามต้องการ แล้วคลิก Save



The image shows the 'Save Game' dialog box in Roblox. It has a dark theme with a blue header bar. The 'Basic Info' tab is selected. The form contains the following fields and options:

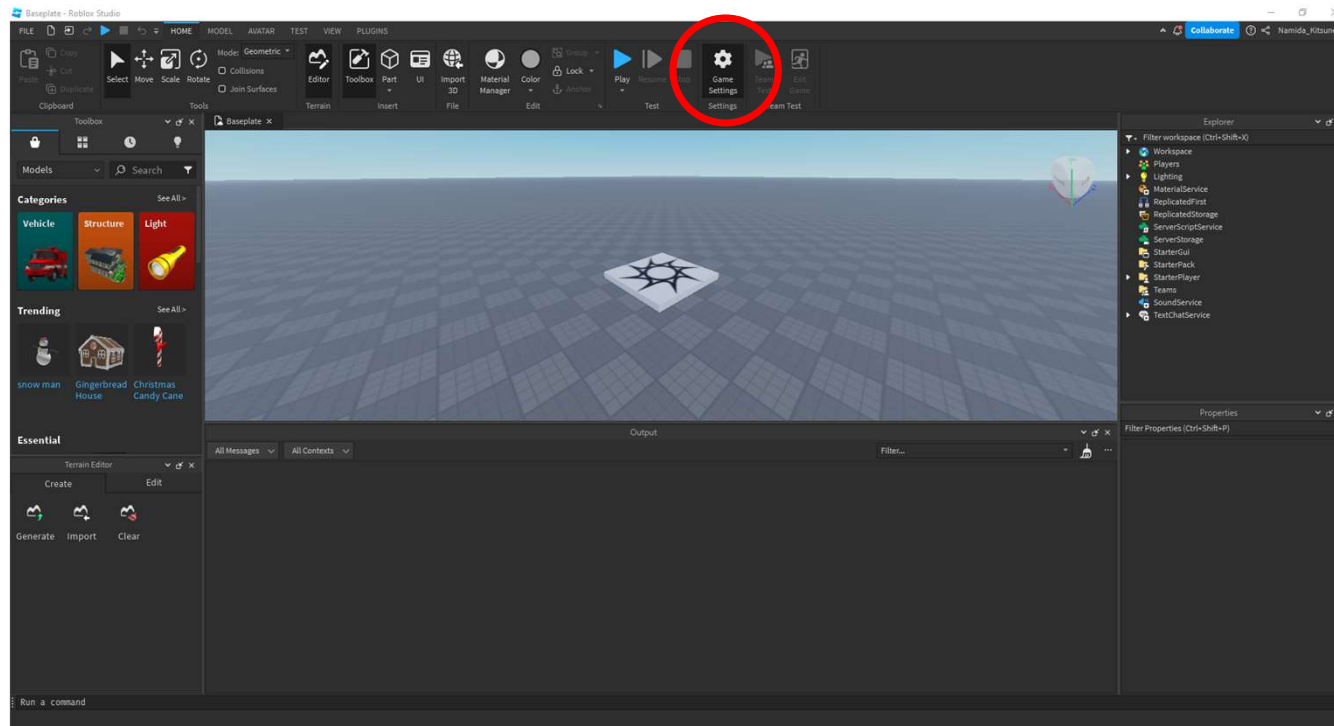
- Name:** A text input field containing 'Untitled Game' with a character count of 13/50.
- Description:** A text area with a character count of 0/1000.
- Creator:** A dropdown menu set to 'Me'.
- Genre:** A dropdown menu set to 'All'.
- Devices:** A group of checkboxes for 'Computer', 'Phone', 'Tablet', and 'VR', all of which are checked. The 'Console' checkbox is unchecked.
- Team Create:** A toggle switch that is turned on (green). Below it, text reads 'Enables collaboration and autosave to cloud' with a 'Learn more' link.

At the bottom of the dialog, there are three buttons: 'Update existing game...' (disabled), 'Cancel', and 'Save'. The 'Save' button is highlighted with a red rectangular border.

วิธีเปิดใช้งานคำขอ HTTP วิธีที่หนึ่ง

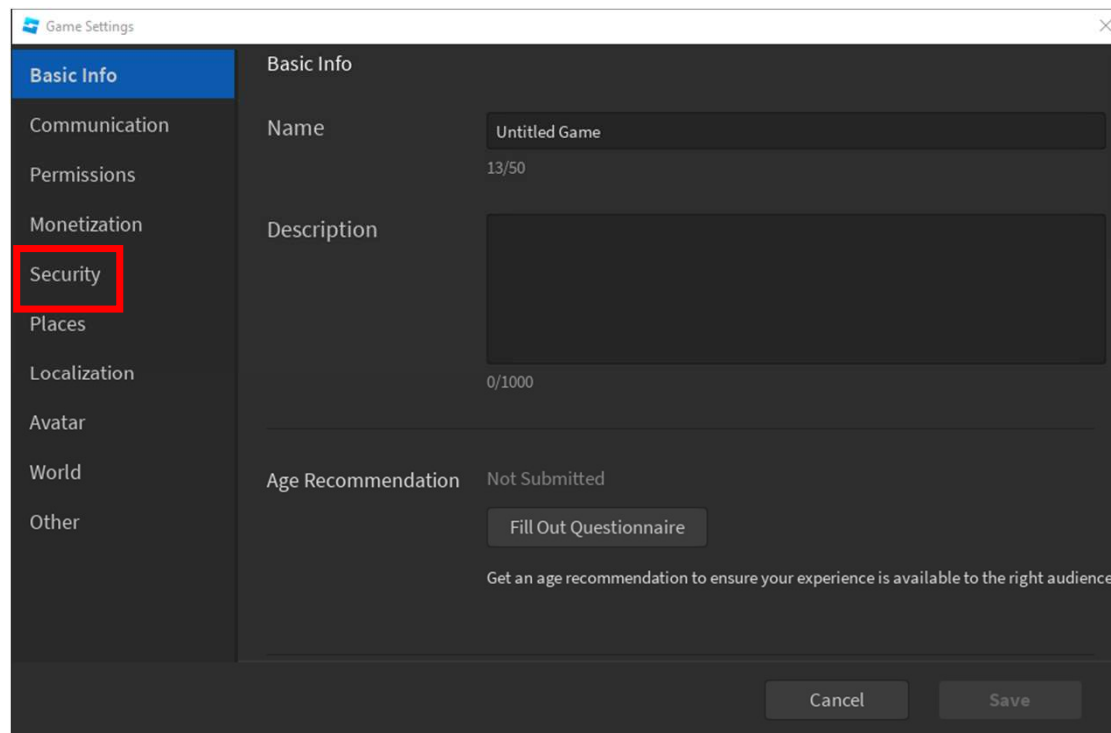
วิธีเปิดใช้งานคำขอ HTTP วิธีที่หนึ่ง

คลิก Game Settings



วิธีเปิดใช้งานคำขอ HTTP วิธีที่หนึ่ง (ต่อ)

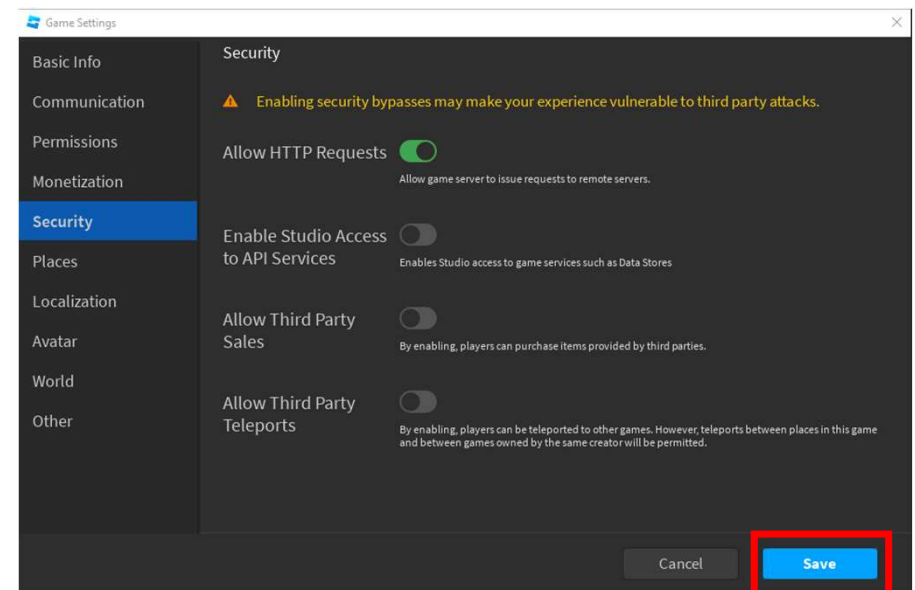
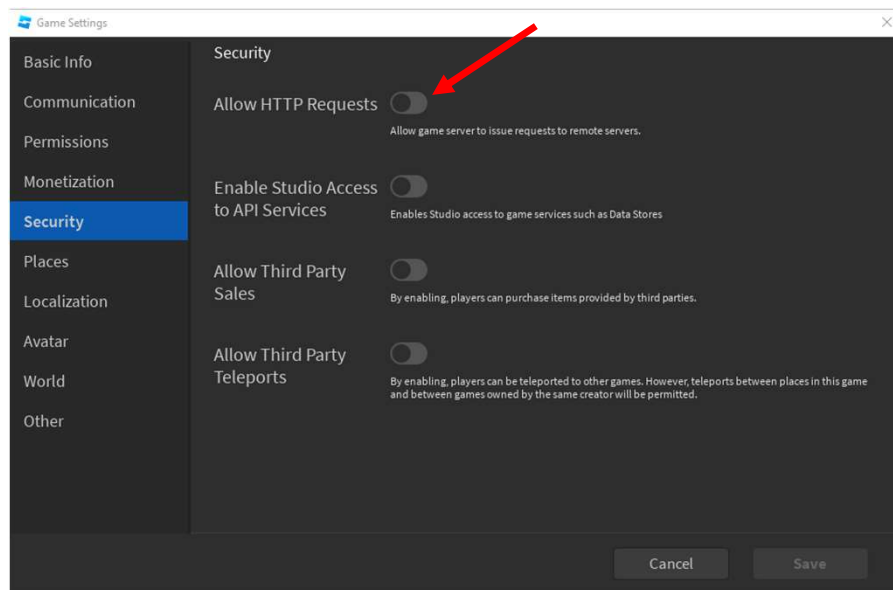
คลิก Security



The screenshot shows the 'Game Settings' window. On the left sidebar, the 'Security' tab is highlighted with a red rectangle. The main panel displays the 'Basic Info' section, which includes a 'Name' field with the text 'Untitled Game' and a character count of '13/50'. Below this is a 'Description' field with a character count of '0/1000'. At the bottom of the main panel, there is an 'Age Recommendation' section showing 'Not Submitted' and a 'Fill Out Questionnaire' button. A note at the bottom of the window states: 'Get an age recommendation to ensure your experience is available to the right audience'. At the very bottom of the window are 'Cancel' and 'Save' buttons.

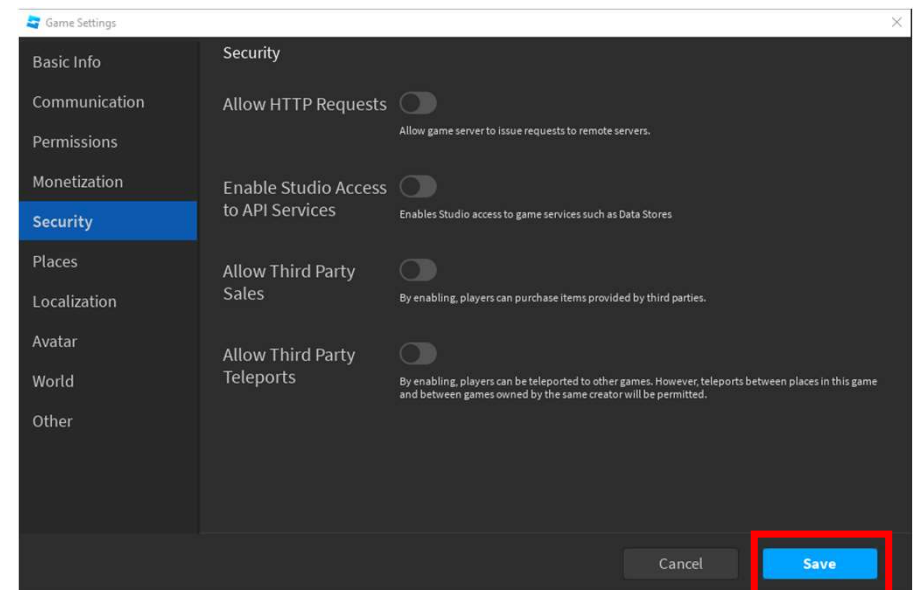
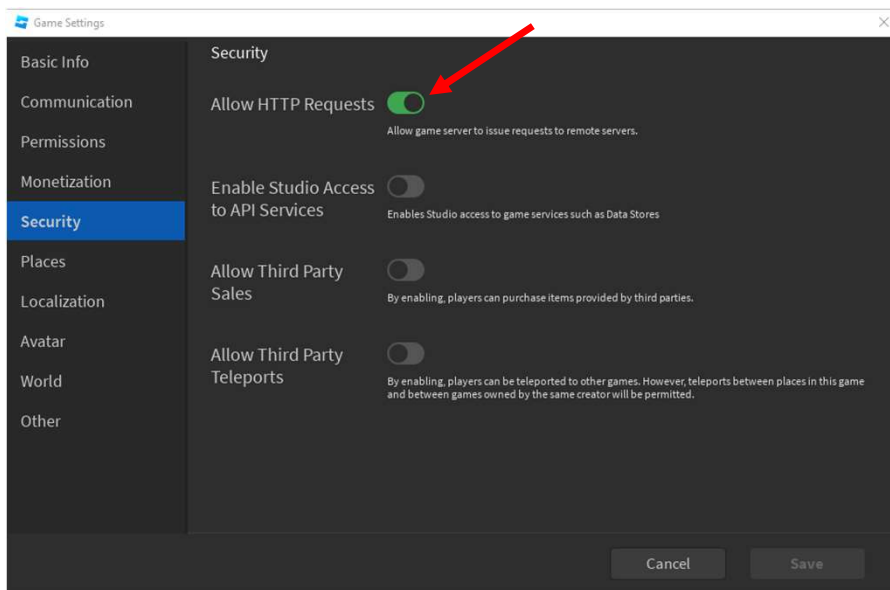
วิธีเปิดใช้งานคำขอ HTTP วิธีที่หนึ่ง (ต่อ)

คลิก Allow HTTP Requests เพื่อเปิดใช้งานคำขอ HTTP > คลิก Save



วิธีปิดใช้งานคำขอ HTTP วิธีที่หนึ่ง (ต่อ)

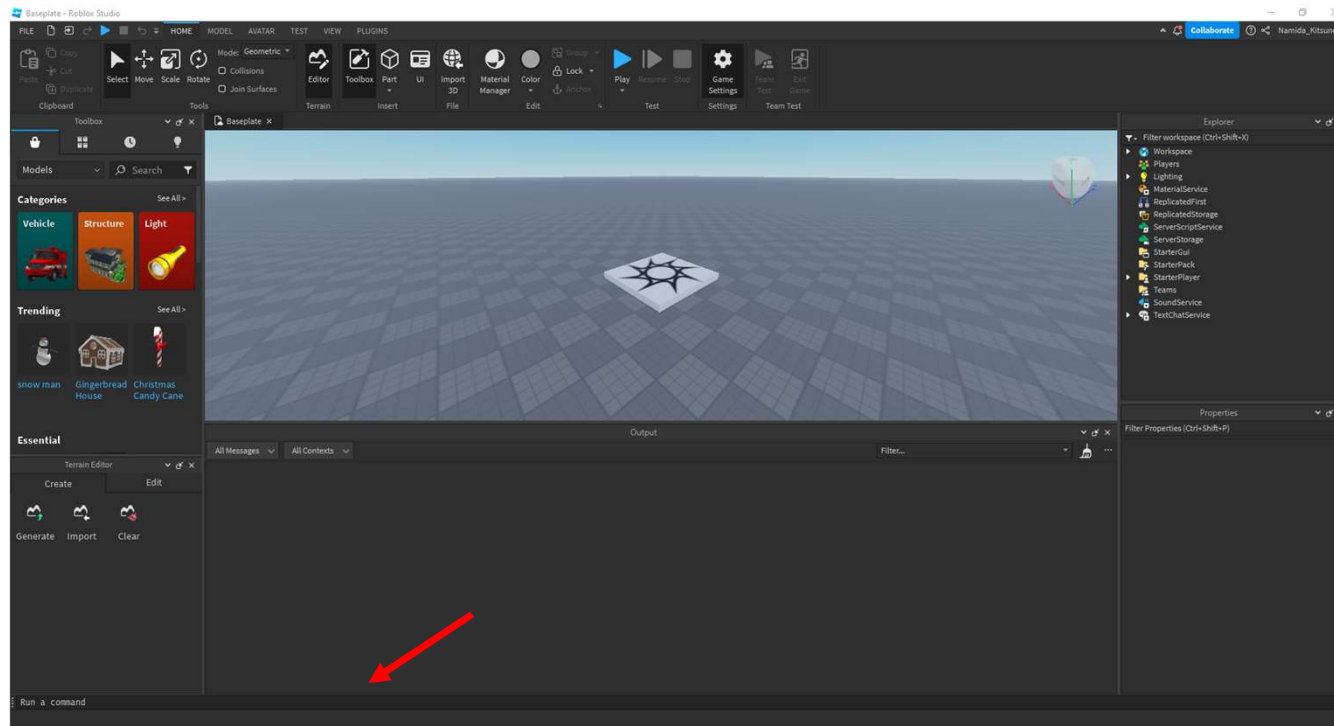
คลิก Allow HTTP Requests เพื่อปิดใช้งานคำขอ HTTP > คลิก Save



วิธีเปิดใช้งานคำขอ HTTP วิธีที่สอง
(วิธีนี้ไม่สามารถใช้ได้ขณะที่เกมทำงาน)

วิธีเปิดใช้งานคำขอ HTTP วิธีที่สอง

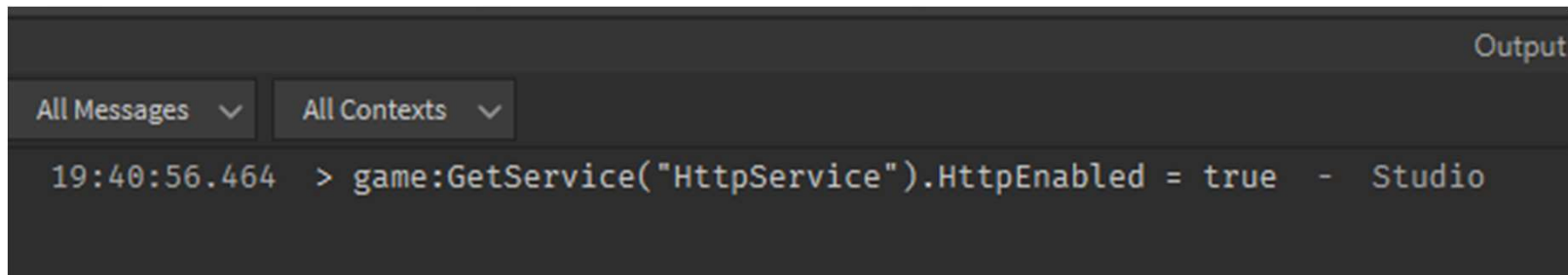
คลิก Command Bar



วิธีเปิดใช้งานคำขอ HTTP วิธีที่สอง (ต่อ)

พิมพ์ `game:GetService("HttpService").HttpEnabled = true` >

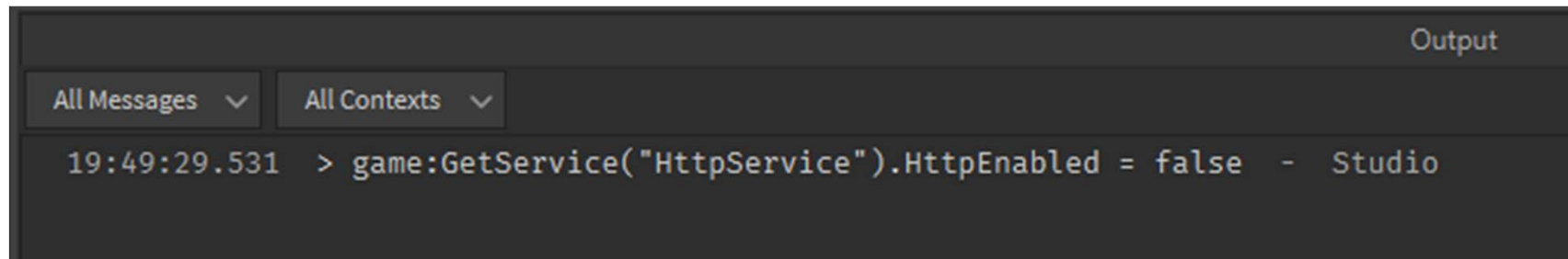
กดปุ่ม Enter ผลลัพธ์ที่หน้าต่างส่งออก (Output) ควรมีข้อความขึ้นดังภาพ



The screenshot shows the 'Output' window of a development studio. At the top right, the word 'Output' is displayed. Below it, there are two dropdown menus: 'All Messages' and 'All Contexts', both with downward-pointing arrows. The main area of the window displays a log entry: '19:40:56.464 > game:GetService("HttpService").HttpEnabled = true - Studio'. The text is white on a dark background.

วิธีปิดใช้งานคำขอ HTTP วิธีที่สอง (ต่อ)

พิมพ์ `game:GetService("HttpService").HttpEnabled = false` >
กดปุ่ม Enter ผลลัพธ์ที่หน้าต่างส่งออก (Output) ควรมีข้อความขึ้นดังภาพ



The screenshot shows the 'Output' window in Roblox Studio. At the top right is the label 'Output'. Below it are two dropdown menus: 'All Messages' and 'All Contexts'. The main area displays a log entry: '19:49:29.531 > game:GetService("HttpService").HttpEnabled = false - Studio'.

```
19:49:29.531 > game:GetService("HttpService").HttpEnabled = false - Studio
```

ทำความรู้จักเมทอดที่มีใน HttpService

เมทอดคืออะไร

เมทอด (method) เป็นคำที่ใช้ในภาษาอังกฤษ ซึ่งมีอีกคำแปลหนึ่งว่า วิธี (method) โดยหมายถึง ทำนองหรือหนทางที่จะทำ เช่น ถ้ากล่าวว่า walking เป็น method จะสามารถแปลได้ว่า วิธีการเดิน เช่นกัน

เคล็ดลับ เต็มคำว่า “วิธี” ไว้หน้าทุกสิ่งที่เป็นเมทอด อาจทำให้ทราบได้ว่า เมทอดนั้นทำงานอย่างไร

เมทอดที่มีใน HttpService

- GenerateGUID (วิธีก่อกำเนิด GUID: Globally Unique Identifier)
- **GetSecret (วิธี Get Secret)**
- JSONDecode (วิธีถอดรหัส JSON: JavaScript Object Notation)
- JSONEncode (วิธีเข้ารหัส JSON: JavaScript Object Notation)
- UriEncode (วิธีเข้ารหัส URL: Uniform Resource Locator)
- GetAsync (วิธี Get Asynchronous)
- PostAsync (วิธี Post Asynchronous)
- RequestAsync (วิธี Request Asynchronous)

UUID และ GUID คืออะไร

Universally Unique Identifier (UUID) คือวิธีการมาตรฐานที่ใช้สำหรับสร้างค่าที่ไม่ซ้ำกัน แม้ว่าความน่าจะเป็นที่ UUID จะถูกทำซ้ำนั้นไม่ใช่ศูนย์ แต่โดยทั่วไปถือว่าใกล้เคียงกับศูนย์มากพอที่จะละเลยได้

Globally Unique Identifier (GUID) เป็นค่าที่สร้างมาจาก UUID โดย Microsoft และใช้คำนี้ในระบบปฏิบัติการ Windows

เมทอด GenerateGUID

เมทอด GenerateGUID มีหน้าที่สุ่มสร้างสายอักขระด้วยวิธีการ
Universally Unique Identifier (UUID)

หลักการในการสุ่ม คือ ให้เลขโดดฐานสิบหกจำนวน 32 ตัว เป็นตัวแทนของ
UUID ซึ่งการแสดงผลจะแบ่งเป็น 5 กลุ่ม และแยกกันได้โดยเครื่องหมายปิดสั้น
(hyphen) ในรูปแบบ 8-4-4-4-12 ซึ่งผลรวมคือ 36 อักขระ

เมทอด GenerateGUID (ต่อ)

รูปแบบของการสุ่ม

8-4-4-4-12

ผลลัพธ์ของการสุ่ม

123e4567-e89b-12d3-a456-426655440000

เมทอด GenerateGUID (ต่อ)

พารามิเตอร์ (Parameters)	รายละเอียด	ค่าเริ่มต้น
wrapInCurlyBraces: boolean	มีไว้กำหนดว่าการส่งคืนสายอักขระ ควรห่อหุ้มด้วย {วงเล็บปีกกา} ไหม	true
ส่งคืน (Returns)	รายละเอียด	
string	UUID ที่สร้างขึ้นแบบสุ่ม	

เมท็อด JSONDecode

เมท็อด JSONDecode มีหน้าที่แปลง JSON string ให้เป็น Lua table

ความรู้เพิ่มเติม สามารถใช้งาน เมท็อด JSONDecode แม้ว่าจะไม่ได้เปิดใช้งาน

คำขอ HTTP

เมทอด JSONDecode (ต่อ)

พารามิเตอร์ (Parameters)	รายละเอียด	ค่าเริ่มต้น
input: string	JSON string ที่ต้องการถอดรหัส	ไม่มี
ส่งคืน (Returns)	รายละเอียด	
Variant	JSON ในรูปแบบ Lua table	

เมท็อด JSONEncode

เมท็อด JSONEncode มีหน้าที่แปลง Lua table ให้เป็น JSON string โดย key ใน Lua table ที่มี value = nil จะไม่ถูกสร้างขึ้น

ความรู้เพิ่มเติม สามารถใช้งาน เมท็อด JSONEncode แม้ว่าจะไม่ได้เปิดใช้งาน คำขอ HTTP

เมทอด JSONEncode (ต่อ)

พารามิเตอร์ (Parameters)	รายละเอียด	ค่าเริ่มต้น
input: Variant	JSON ในรูปแบบ Lua table ที่ต้องการเข้ารหัส	ไม่มี
ส่งคืน (Returns)	รายละเอียด	
string	JSON string	

เมทอด UriEncode

เมทอด UriEncode มีหน้าที่ **percent-encoding** คือ การนำอักขระที่ **สงวน**ไว้มาแทนที่ด้วยเครื่องหมายเปอร์เซ็นต์ % และเลขโดดฐานสิบหก 2 ตัว เพื่อจัดรูปแบบ URL สำหรับใช้กับ เมทอด GetAsync / PostAsync หรือ ส่งคำขอ HTTP เป็นชนิดข้อมูล ApplicationUrlEncoded

เมทอด UrlEncode (ต่อ)

ตัวอย่าง

[`https://www.roblox.com/discover#/`](https://www.roblox.com/discover#/)

ถูกแทนที่เป็น

`https%3A%2F%2Fwww%2Eroblox%2Ecom%2Fdiscover%23%2F`

เมทอด UrlEncode (ต่อ)

พารามิเตอร์ (Parameters)	รายละเอียด	ค่าเริ่มต้น
input: string	string ที่ต้องการเข้ารหัส	ไม่มี
ส่งคืน (Returns)	รายละเอียด	
string	string ที่เข้ารหัสแล้ว	

เมทอด GetAsync

เมทอด `GetAsync` มีหน้าที่ ส่งคำขอ HTTP GET โดยเมทอด `GetAsync` ทำงานคล้ายกับเมทอด `RequestAsync` ยกเว้นเรื่องการรับพารามิเตอร์ และการส่งคืนค่า เนื่องจาก เมทอด `GetAsync` รับพารามิเตอร์แบบที่ละพารามิเตอร์ และส่งคืนค่ามาเพียง `body` เท่านั้น

เคล็ดลับ ประโยชน์ของเมทอด `GetAsync` คือใช้ในการเขียน shorthand เท่านั้น ในกรณีส่วนมากควรใช้เมทอด `RequestAsync`

เมทอด GetAsync (ต่อ)

พารามิเตอร์ (Parameters)	รายละเอียด	ค่าเริ่มต้น
url: string	ที่อยู่เว็บที่เราขอข้อมูล	ไม่มี
nocache: boolean	ต้องการเก็บผลลัพธ์จากการเรียกใช้เมทอด GetAsync ครั้งก่อนหน้าไหม (เรียกสิ่งนี้ว่าการเก็บ cache)	false
headers: Variant	ใช้เพื่อระบุ HttpRequestType	ไม่มี
ส่งคืน (Returns)	รายละเอียด	
string	body ที่ได้จากการตอบสนอง โดยคำขอ GET	

เมทอด GetAsync (ต่อ)

ตัวอย่างรูปแบบและการทำงานจริง

รูปแบบ

```
HttpService.GetAsync(url, nocache, headers)
```

การทำงานจริง

```
HttpService.GetAsync("http://api.open-notify.org/astros.json",  
true, [{"HttpRequestType"} = "Default"])
```

เคล็ดลับ ควรใช้งาน **pcall (protected call)** เพื่อตรวจสอบความถูกต้อง

เมทอด PostAsync

เมทอด `PostAsync` มีหน้าที่ ส่งคำขอ HTTP POST โดยเมทอด `PostAsync` ทำงานคล้ายกับเมทอด `RequestAsync` ยกเว้นเรื่องการรับพารามิเตอร์ และการส่งคืนค่า เนื่องจาก เมทอด `PostAsync` รับพารามิเตอร์แบบที่ละพารามิเตอร์ และส่งคืนค่ามาเพียง `body` เท่านั้น

เคล็ดลับ ประโยชน์ของเมทอด `PostAsync` คือใช้ในการเขียน shorthand เท่านั้น ในกรณีส่วนมากควรใช้เมทอด `RequestAsync`

เมทอด PostAsync (ต่อ)

พารามิเตอร์ (Parameters)	รายละเอียด	ค่าเริ่มต้น
url: string	เว็บปลายทางที่เราส่งข้อมูล	ไม่มี
data: string	ข้อมูลที่ต้องการส่ง	ไม่มี
content_type: HttpContentType	ใช้เพื่อระบุ HttpContentType ว่าข้อมูลที่ต้องการส่งอยู่ในรูปแบบไหน	ApplicationJson
compress: boolean	เป็นการกำหนดว่าต้องการให้ข้อมูลถูกบีบอัด (gzipped) ก่อนส่งไปเว็บปลายทางไหม	false
headers: Variant	ใช้เพื่อระบุ HttpRequestType	ไม่มี
ส่งคืน (Returns)	รายละเอียด	
string	body ที่ได้จากการตอบสนอง โดยคำขอ POST	

เมทอด PostAsync (ต่อ)

ตัวอย่างรูปแบบและการทำงานจริง

รูปแบบ

```
HttpService.PostAsync(url, data, content_type, compress, headers)
```

การทำงานจริง

```
HttpService.PostAsync("http://localhost:3001/api/added", [{"uuid":"414AAC8D-367E-4064-ADF3-73F49A1AF699","username":"Namida_Kitsune"}],  
Enum.HttpContentType.ApplicationJson, false, [{"HttpRequestType"] = "Default"})
```

เคล็ดลับ ควรใช้งาน **pcall (protected call)** เพื่อตรวจสอบความถูกต้อง

เมทอด RequestAsync

เมทอด RequestAsync มีหน้าที่ ส่งคำขอ HTTP โดยส่วนใหญ่มักจะเป็น GET หรือ POST ซึ่งเมทอด RequestAsync จะใช้ **table** เพื่อระบุข้อมูลคำขอ เช่น url, headers เป็นต้น และจะส่งคืนค่ากลับเป็น **table** ที่มีข้อมูลละเอียดมากกว่าเมทอด GetAsync และเมทอด PostAsync

ความรู้เพิ่มเติม table สำหรับภาษา luau ใน roblox studio แยกได้อีกสองชื่อตามวิธีการเขียน คือ array และ dictionary เราจึงสามารถเรียกรวม ๆ ได้ว่าเป็น table

เมทอด RequestAsync (ต่อ)

พารามิเตอร์ (Parameters)	รายละเอียด	ค่าเริ่มต้น
requestOptions: table	table ที่มีข้อมูลที่ทางฝั่ง server ร้องขอ	ไม่มี
ส่งคืน (Returns)	รายละเอียด	
table	table ที่ได้จากการตอบสนอง	

เมทอด RequestAsync (ต่อ)

ข้อมูลที่ใช้ในพารามิเตอร์ requestOptions

ชื่อ	ชนิด	ที่จำเป็นต้องใส่	คำอธิบาย
Url	String	ใช่	เว็บปลายทางที่เราขอหรือส่งข้อมูล โดยต้องใช้โปรโตคอล http หรือ https
Method	String	ไม่	วิธีการที่ใช้ส่งคำขอ มักเป็น GET หรือ POST
Headers	table	ไม่	ใช้เพื่อระบุ Header ส่วนใหญ่ HTTP Headers ที่กำหนดไม่ได้ ตัวอย่างเช่น Content-Length, User-Agent และ Roblox-Id เป็นต้น
Body	String	ไม่	ข้อมูลที่ต้องการส่ง โดยสามารถส่งเป็นสายอักขระ หรือข้อมูลชนิดไบนารีก็ได้ ยกเว้นหากตั้ง Method เป็น GET อาจจำเป็นต้องระบุ Content-Type เมื่อส่งข้อมูลรูปแบบ JSON หรือรูปแบบอื่น

เมทอด RequestAsync (ต่อ)

ผลลัพธ์ของการใช้งานเมทอด RequestAsync

ชื่อ	ชนิด	คำอธิบาย
Success	Boolean	สถานะความสำเร็จของคำขอ สิ่งนี้จะเกิดขึ้นก็ต่อเมื่อ StatusCode อยู่ในช่วง 200 ถึง 299
StatusCode	Integer	code ที่ใช้ระบุ status ของการตอบสนอง
StatusMessage	String	ข้อความ Status ที่ถูกส่งกลับ
Headers	Dictionary	HttpRequestType ที่ตั้งไว้สำหรับการตอบสนอง
Body	ไม่มี	Body ที่ได้จากการตอบสนอง

แหล่งความรู้ในการศึกษาเพิ่มเติม

- HttpService [คลิกที่นี่](#)
- HttpCachePolicy [คลิกที่นี่](#)
- HttpContentType [คลิกที่นี่](#)
- HttpError [คลิกที่นี่](#)
- HttpRequestType [คลิกที่นี่](#)
- SourceCode HttpService [คลิกที่นี่](#)