

# Solidity Challenge

## Introduction

The purpose of this challenge is to expose you to the Safe contracts, get a feeling on how you structure your solidity code and see how you set up your contract development environment.

For this challenge we expect that you write a [Safe Module](#). A Safe Module allows you to execute transactions via the Safe contract with an alternative access scheme.

## Objective

The contract written for this challenge should use this functionality to allow accounts that are not related to the Safe to withdraw a predetermined amount of a specific token.

To better understand this let's take an example:

Alice has set up a Safe which they use to hold all of their Unicorn tokens. They want to hand out unicorn tokens easily without having Alice trigger a transaction each time. Therefore they enable the module written in this challenge. Now they can generate a signature which allows anyone to withdraw Unicorn tokens from their Safe. With this they generate a signature for a withdrawal of 10 Unicorn tokens and share it with Bob. Now Bob can use this signature and interact with the module which then will send 10 Unicorn tokens from Alice Safe to Bob's address.

If there are any questions on this challenge feel free to reach out to the hiring team and we will get back to you as soon as possible.

## Features

The following features should be included in the module:

- Module should be specific to a token address and to a Safe
  - The module is not required to work for multiple token address or multiple safe addresses
- Method to withdraw tokens (via a transaction)
  - The method should take the
    - **amount of tokens to withdraw**
    - **a beneficiary (receiver of the tokens)**
    - **signature of an owner of the Safe it is attached to**

Optional features:

- Signatures should expire after a set time
- The module should require the signatures of multiple owners (to meet the threshold requirement of the attached Safe)

## Expectations on Code Quality

- Tests for all smart contract functionality
- NatSpec for public methods
- Code comments on relevant code sections

## Tools

- Hardhat or Foundry

## Safe code references:

- Enable Module:  
<https://github.com/safe-global/safe-contracts/blob/main/contracts/base/ModuleManager.sol#L32>
- Execute Transaction from a Module:  
<https://github.com/safe-global/safe-contracts/blob/main/contracts/base/ModuleManager.sol#L61>
- Check owner signatures:  
<https://github.com/safe-global/safe-contracts/blob/main/contracts/GnosisSafe.sol#L233-L245>
- Getting started with modules:  
<https://gist.github.com/rmeissner/c66371a4b030109cde1020f427122462>
- Example Module:  
<https://gist.github.com/rmeissner/208e35db13d0ebf53b6e35838f1c8122>