# ChainSafe Gaming SDK

## Minter - Beta EVM Testnet

This minter is currently in beta and works only on Ethereum testnets. i.e. Goerli/Rinkeby/Kovan/Ropsten

### Mint NFT via Private Key

This implementation will mint an NFT via a private key.

```
CreateMintModel.Response nftResponse = await EVM.CreateMint(chain, network, account, to, c
if (nftResponse != null)
    {
        string chainId = await EVM.ChainId(chain, network, "");
        // private key of account
        string privateKey = "ADD_YOUR_PRIVATE_KEY";
        string transaction = await EVM.CreateTransaction(chain, network, nftResponse.tx.ac
        nftResponse.tx.to, nftResponse.tx.value, nftResponse.tx.data, nftResponse.tx.gasPr
        string signature = Web3PrivateKey.SignTransaction(privateKey, transaction, chainId
        string responseBroadcast = await EVM.BroadcastTransaction(chain, network, nftRespo
        nftResponse.tx.gasPrice, nftResponse.tx.gasLimit, "");
    }
```

### Mint NFT via Web3Wallet

This implementation will mint an NFT via Web3Wallet.

```
CreateMintModel.Response nftResponse = await EVM.CreateMint(chain, network, account, to, 
account = PlayerPrefs.GetString("Account");
// connects to user's browser wallet (metamask) to send a transaction
  try
  {
     string response = await Web3Wallet.SendTransaction(chainId, nftResponse.tx.to, nftRes
     Debug.Log(response);
     } catch (Exception e) {
     Debug.LogException(e, this);
  }
```

### Mint NFT via WebGL

This implementation will mint an NFT via WebGL.

```
CreateMintModel.Response nftResponse = await EVM.CreateMint(chain, network, account, to, c
// connects to user's browser wallet (metamask) to send a transaction
try
```

```
{
    string response = await Web3GL.SendTransactionData(nftResponse.tx.to, nftResponse.tx.va
    print("Response: " + response);
    } catch (Exception e) {
    Debug.LogException(e, this);
}
```