

# jsHost Documentation

The **jsHost** class provides a hosting environment that allows you to execute JavaScript code and access various functionalities and classes from C#. This documentation outlines the capabilities of the **jsHost** and provides details about the available classes and functions that can be accessed from JavaScript.

## Preprocessor

The **jsHost** includes a preprocessor that allows you to define macros using the **#define** directive. Macros can be used to simplify and customize your JavaScript code. They can be defined at the beginning of the script and will be replaced with their corresponding values during preprocessing.

Example:

```
#define PI 3.14159
var radius = 5;
var area = PI * radius * radius;
Console.WriteLine("The area of the circle is: " + area);
```

In the above example, the macro **PI** is defined with the value **3.14159**. During preprocessing, all occurrences of **PI** in the script will be replaced with **3.14159**, resulting in the correct calculation of the circle's area.

## Available Classes and Functions

The **jsHost** exposes various C# classes and their corresponding functions that can be accessed from JavaScript. Below is a list of the available classes and their functions:

### nFinance Class

The **nFinance** class provides functions related to the nFinance Accounting Application.

Functions:

- **FetchData(data: string, key: string): string**: Retrieves a value from a given data string based on the specified key.
- **UpdateData(data: string, key: string, newValue: string): string**: Updates the value of a specified key in the given data string and returns the updated string.
- **FetchSZ(data: string): number**: Returns the count of key-value pairs in the given data string.

- **CountWords(input: string, del: string): number:** Returns the number of words in the input string using the specified delimiter (default delimiter is '~').
- **GetDatabaseFolderPath(): string:** Returns the path of the database folder.
- **GetCustomerCount(): number:** Returns the count of customers.
- **GetOfferCount(): number:** Returns the count of offers.
- **GetAllCustomers(): string[]:** Returns an array containing all customer IDs.
- **GetAllOffers(): string[]:** Returns an array containing all offer IDs.
- **GetItemCount(): number:** Returns the count of items.
- **GetAllItemNames(): string[]:** Returns an array containing all item names.
- **GetCustomerNCF(id: string): string:** Returns the content of the customer's info file specified by the ID.
- **GetOfferNCF(id: string): string:** Returns the content of the offer's info file specified by the ID.
- **GetOfferText(id: string, full: boolean): string:** Returns the content of the offer text file specified by the ID. If **full** is **true**, it returns the full offer text file, otherwise, it returns the abbreviated offer text file.
- **GetItemAvailability(id: string): number:** Returns the availability of an item specified by the ID.
- **GetExpenseIDsForYear(year: number): string[]:** Returns an array containing the expense IDs for the specified year.
- **GetExpenseNCF(id: string): string:** Returns the content of the expense's info file specified by the ID.
- **GetUserSettings(): string:** Returns the content of the user settings file.

## Example usage:

```
var nfinance = new nFinance("%PathToFinanceDBFolder%");
// Accessing functions
var customerCount = nfinance.GetCustomerCount();
Console.WriteLine("Number of customers: " + customerCount);
```

```
var itemAvailability = nfinance.GetItemAvailability("MyArticle");  
Console.WriteLine("Item availability: " + itemAvailability);  
Console.ReadKey();
```

**Note:** This documentation covers only the available functions from the **nFinance** class. For a complete understanding of the capabilities of the **jsHost**, refer to the provided code and explore other available classes and their functions.