

**NAME**

MicroConfig - miniaturní knihovna pro načítání a vytváření jednoduchých konfiguračních souborů.

**DESCRIPTION**

**MicroConfig** je jednoduchá knihovna pro vytváření konfiguračních souborů. Knihovna byla napsána v jazyce C# pro platformu .NET a není závislá na žádné vnější knihovně (samozřejmě s výjimkou standardní knihovny). Pomocí reflexe iterativně prochází členy tříd rozšiřující třídu Config a ukládá jednoduché veřejné členy do stejnojmenného souboru ve složce 'configs'.

**REFERENCE****static string Serialize(object)**

Serializuje objekt, parametr object je jeden objekt třídy rozšiřující třídu Config. Vrací textový řetězec serializovaných hodnot.

**static void Save(object)**

Serializuje objekt a uloží jej do konfiguračního souboru.

**static void Save(object, path)**

Serializuje objekt a uloží jej do konfiguračního souboru specifikovaného proměnnou path.

**void Deserialize()**

Načte konfigurační soubor ze složky configs.

**void Deserialize(dirpath)**

Načte konfigurační soubor ze složky specifikované parametrem dirpath.

**Config()**

Konstruktor pro výchozí konfigurační soubor

**Config(bool autoload)**

Konstruktor pro konfigurační soubor s automatickým načítáním.

**MODUS OPERANDI**

- 1 Po vytvoření instance samonačítacího konfiguračního objektu nebo po zavolání metody Deserialize() MicroConfig vyhledá a přečte konfigurační soubor. Pokud neexistuje, vytvoří nový podle výchozích hodnot členů dané třídy.
- 2 Text je poté rozdělen na řádky a prázdné řádky jsou vymazány.
- 3 Každý řádek je rozebrán a vytvoří se z něj dvojice klíč:hodnota.
- 4 Pomocí reflexe se získají informace o typu objektu a jeho členech.
- 5 Každému členu, pro který byl nalezen klíč, je nastavena patřičná hodnota a tím je ukončeno načítání.
- 6 Při ukládání je proveden opačný postup. Zavolání funkce Save() načte pomocí reflexe všechny členy dané třídy a vybere z nich ty, které jsou veřejné, primitivních typů a jejichž jméno nezačíná podtržítkem.
- 7 Z vybraných členů jsou vytvořeny dvojice klíč:hodnota.
- 8 Dvojice jsou zapsány do souboru configs/Třída.cfg (pokud to není jinak specifikováno) ve formě 'název = hodnota'. Členy, jejichž hodnota nebyla dříve určena v konfiguračním souboru, mají přidány záznamy s výchozími hodnotami

**LIMITACE**

MicroConfig nedokáže zpracovat složitější typy a pole.

**MAPA KÓDU****Config.cs**

Obsahuje kompletní zdrojový kód knihovny.

**PŘÍKLAD POUŽITÍ**

Jednoduché příklady je možné nalézt u serveru net\_47sb\_59vm. Zde je příklad třídy, ze které lze vytvořit konfigurační soubor:

```
public class Test : Config
{
    public string test1 = "test1";
    public string test2 = "mandarinka";
    public int cislo = 25;
    public bool pravda = true;
    public bool lez = false;
}
```

## LICENCE

MicroConfig je licencován licencí 'ISC License'. Jedná se o krátkou licenci, která je funkčně podobná licenci MIT, ale kratší. Je zároveň preferovanou licencí skupiny suckless.org.

ISC License (ISC)

Copyright 2017 Lukáš Hozda (luk.hozda@gmail.com, github: luciusmagn)

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.