

[chromium](#) / [chromium](#) / [src](#) / [refs/tags/64.0.3282.24](#) / [.](#) / [chrome](#) / [browser](#) / [spellchecker](#) / **spellcheck_factory.cc**

blob: 148ec2086d1f6389dbf32d5a3a1e5bc81c19a46d [[file](#)]

```
1 // Copyright (c) 2012 The Chromium Authors. All rights reserved.
2 // Use of this source code is governed by a BSD-style license that can be
3 // found in the LICENSE file.
4
5 #include "chrome/browser/spellchecker/spellcheck_factory.h"
6
7 #include "base/memory/ptr_util.h"
8 #include "chrome/browser/profiles/incognito_helpers.h"
9 #include "chrome/browser/spellchecker/spellcheck_service.h"
10 #include "chrome/grit/locale_settings.h"
11 #include "components/keyed_service/content/browser_context_dependency_manager.h"
12 #include "components/pref_registry/pref_registry_syncable.h"
13 #include "components/prefs/pref_service.h"
14 #include "components/spellcheck/browser/pref_names.h"
15 #include "components/user_prefs/user_prefs.h"
16 #include "content/public/browser/browser_context.h"
17 #include "content/public/browser/render_process_host.h"
18 #include "services/service_manager/public/cpp/identity.h"
19 #include "ui/base/l10n/l10n_util.h"
20
21 // static
22 SpellcheckService* SpellcheckServiceFactory::GetForContext(
23     content::BrowserContext* context) {
24     return static_cast<SpellcheckService*>(
25         GetInstance()->GetServiceForBrowserContext(context, true));
26 }
27
28 // static
29 SpellcheckService* SpellcheckServiceFactory::GetForRenderer(
30     const service_manager::Identity& renderer_identity) {
31     content::BrowserContext* context =
32         content::BrowserContext::GetBrowserContextForServiceUserId(
33             renderer_identity.user_id());
34     if (!context)
35         return nullptr;
36     return GetForContext(context);
37 }
38
39 // static
40 SpellcheckServiceFactory* SpellcheckServiceFactory::GetInstance() {
```

```

41     return base::Singleton<SpellcheckServiceFactory>::get();
42 }
43
44 SpellcheckServiceFactory::SpellcheckServiceFactory()
45     : BrowserContextKeyedServiceFactory(
46         "SpellcheckService",
47         BrowserContextDependencyManager::GetInstance()) {
48     // TODO(erg): Uncomment these as they are initialized.
49     // DependsOn(RequestContextFactory::GetInstance());
50 }
51
52 SpellcheckServiceFactory::~SpellcheckServiceFactory() {}
53
54 KeyedService* SpellcheckServiceFactory::BuildServiceInstanceFor(
55     content::BrowserContext* context) const {
56     // Many variables are initialized from the |context| in the SpellcheckService.
57     SpellcheckService* spellcheck = new SpellcheckService(context);
58
59     PrefService* prefs = user_prefs::UserPrefs::Get(context);
60     DCHECK(prefs);
61
62     // Instantiates Metrics object for spellchecking for use.
63     spellcheck->StartRecordingMetrics(
64         prefs->GetBoolean(spellcheck::prefs::kEnableSpellcheck));
65
66     return spellcheck;
67 }
68
69 void SpellcheckServiceFactory::RegisterProfilePrefs(
70     user_prefs::PrefRegistrySyncable* user_prefs) {
71     user_prefs->RegisterListPref(spellcheck::prefs::kSpellCheckDictionaries,
72                                 base::MakeUnique<base::ListValue>());
73     // Continue registering kSpellCheckDictionary for preference migration.
74     // TODO(estade): remove: crbug.com/751275
75     user_prefs->RegisterStringPref(
76         spellcheck::prefs::kSpellCheckDictionary,
77         l10n_util::GetStringUTF8(IDS_SPELLCHECK_DICTIONARY));
78     user_prefs->RegisterBooleanPref(
79         spellcheck::prefs::kSpellCheckUseSpellingService, false);
80 #if defined(OS_ANDROID)
81     uint32_t flags = PrefRegistry::NO_REGISTRATION_FLAGS;
82 #else
83     uint32_t flags = user_prefs::PrefRegistrySyncable::SYNCABLE_PREF;
84 #endif
85     user_prefs->RegisterBooleanPref(
86         spellcheck::prefs::kEnableSpellcheck, true, flags);
87 }
88
89 content::BrowserContext* SpellcheckServiceFactory::GetBrowserContextToUse(

```

```
90     content::BrowserContext* context) const {
91     return chrome::GetBrowserContextRedirectedInIncognito(context);
92 }
93
94 bool SpellcheckServiceFactory::ServiceIsNULLWhileTesting() const {
95     return true;
96 }
```