SEP	3
Title	List fields API
Author	Pablo Hoffman
Created	2009-07-21
Status	Obsolete by :ref:`sep-008` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\scrapy-master\sep\[scrapy-master][sep]sep-002.rst, line 7`); *backlink* Unknown interpreted text role "ref".

SEP-002 - List fields API

This page presents different usage scenarios for the new multi-valued field, called !ListField.

Proposed Implementation

```python
#!python
from scrapy.item.fields import BaseField

class ListField(BaseField):
    def __init__(self, field, default=None):
        self._field = field
        super(ListField, self).__init__(default)

    def to_python(self, value):
        if hasattr(value, '__iter__'): # str/unicode not allowed
            return [self._field.to_python(v) for v in value]
        else:
            raise TypeError("Expected iterable, got %s" % type(value).__name__)

    def get_default(self):
        # must return a new copy to avoid unexpected behaviors with mutable defaults
        return list(self._default)
```

Usage Scenarios

Defining a list field

```python
#!python
from scrapy.item.models import Item
from scrapy.item.fields import ListField, TextField, DateField, IntegerField

class Article(Item):
    categories = ListField(TextField)
    dates = ListField(DateField, default=[])
    numbers = ListField(IntegerField, [])
```

Another case of products and variants which highlights the fact that it's important to instantiate !ListField with field instances, not classes:

```python
#!python
from scrapy.item.models import Item
from scrapy.item.fields import ListField, TextField

class Variant(Item):
    name = TextField()

class Product(Variant):
    variants = ListField(ItemField(Variant))
```

Assigning a list field

```python
#!python
i = Article()

i['categories'] = []
i['categories'] = ['politics', 'sport']
i['categories'] = ['test', 1] -> raises TypeError
i['categories'] = asd -> raises TypeError

i['dates'] = []
```

```python
i['dates'] = ['2009-01-01']  # raises TypeError? (depends on TextField)

i['numbers'] = ['1', 2, '3']
i['numbers'] # returns [1, 2, 3]
```

Default values

```python
#!python
i = Article()

i['categories'] # raises KeyError
i.get('categories') # returns None

i['numbers'] # returns []
```

Appending values

```python
#!python
i = Article()

i['categories'] = ['one', 'two']
i['categories'].append(3) # XXX: should this fail?
```

```python
i['dates'] = ['2009-01-01']  # raises TypeError? (depends on TextField)

i['numbers'] = ['1', 2, '3']
i['numbers'] # returns [1, 2, 3]
```

```python
#!python
i = Article()
```