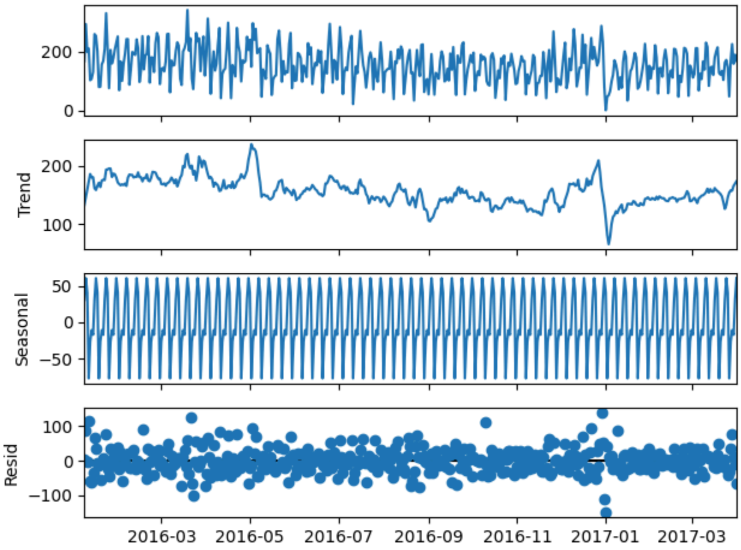
In [62]:

```
train = ts_data_sub[:-21]
test = ts_data_sub[-21:]
```

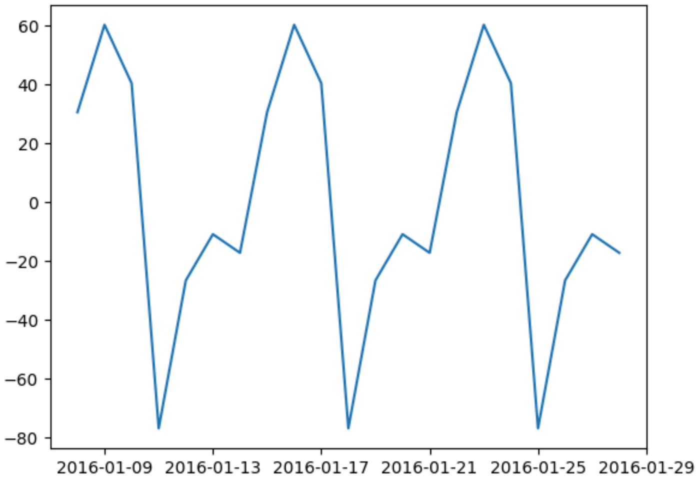
In [63]:

```
# daily_matrix.index.freq = 'D'
train_decomp = seasonal_decompose(train)
train_decomp = seasonal_decompose(x=train, model='additive', period=7, extrapolate_trend=1)
train_decomp.plot();
```



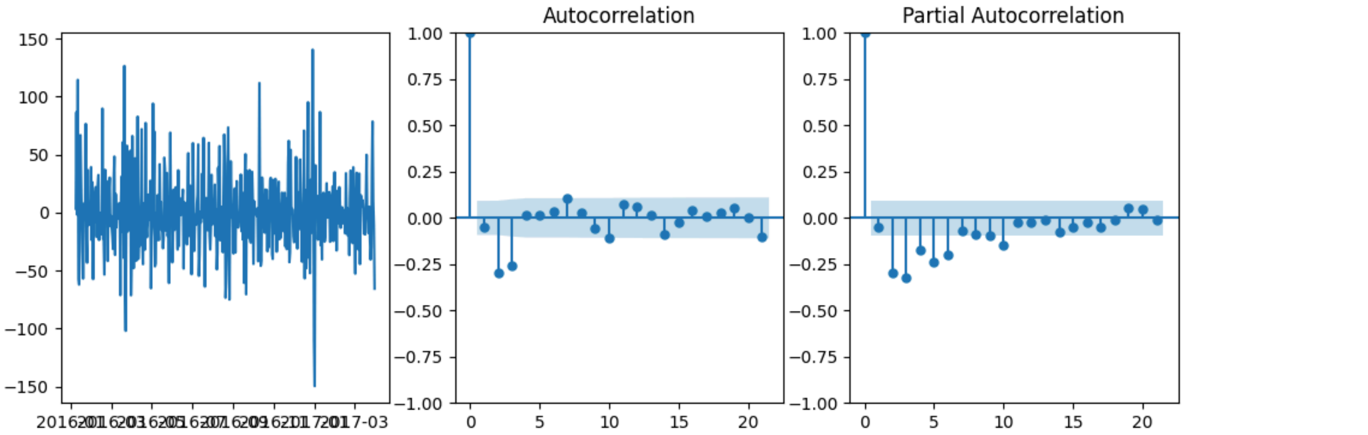
In [64]:

```
plt.plot(train_decomp.seasonal[:21]);
```



In [65]:

```
fig,axs = plt.subplots(1,3,figsize=(12,4))
axs[0].plot(train_decomp.resid.dropna());
plot_acf(train_decomp.resid.dropna(),lags=21,ax=axs[1]);
plot_pacf(train_decomp.resid.dropna(),lags=21,ax=axs[2]);
# ts_data_sub_decomp.resid.index
```

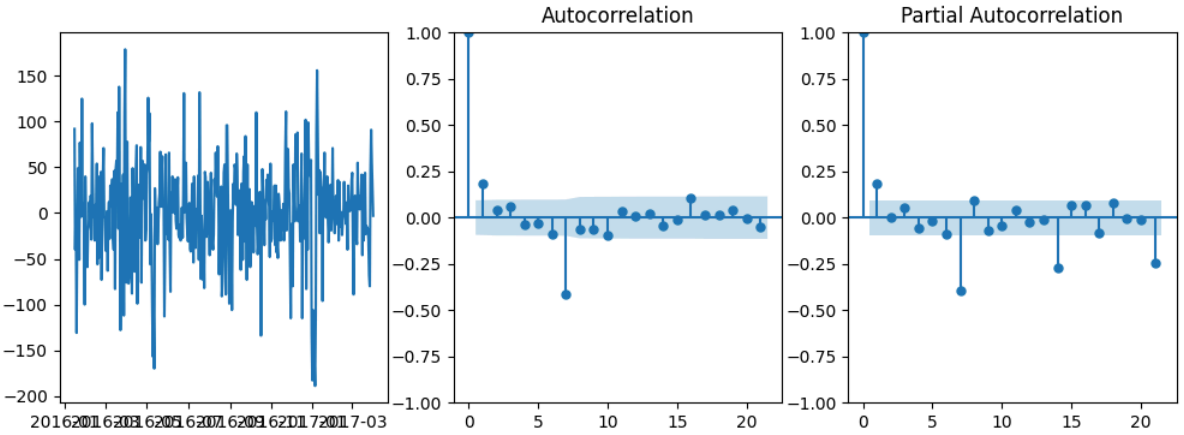


In [ ]:

```
# ts_data_sub_diff = ts_data_sub.diff()
# plt.plot(ts_data_sub_diff);
# plot_acf(ts_data_sub_diff.dropna());
# plot_pacf(ts_data_sub_diff.dropna());
```

In [66]:

```
train_diff7 = train.diff(7).dropna()
fig,axs = plt.subplots(1,3,figsize=(12,4))
axs[0].plot(train_diff7);
plot_acf(train_diff7,lags=21,ax=axs[1]);
plot_pacf(train_diff7,lags=21,ax=axs[2]);
```



In [100…

```
%%time
train_auto = auto_arima(train,seasonal=True,m=7)
```

```
CPU times: total: 32.8 s
Wall time: 37 s
```

In [67]:

```
train_auto.summary()
```

Out[67]:

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	450
Model:	SARIMAX(5, 1, 0)x(2, 0, 0, 7)	**Log Likelihood**	-2378.572
Date:	Wed, 24 Dec 2025	**AIC**	4773.144
Time:	13:13:10	**BIC**	4806.000
Sample:	01-08-2016	**HQIC**	4786.095
	- 04-01-2017		
Covariance Type:	opg		

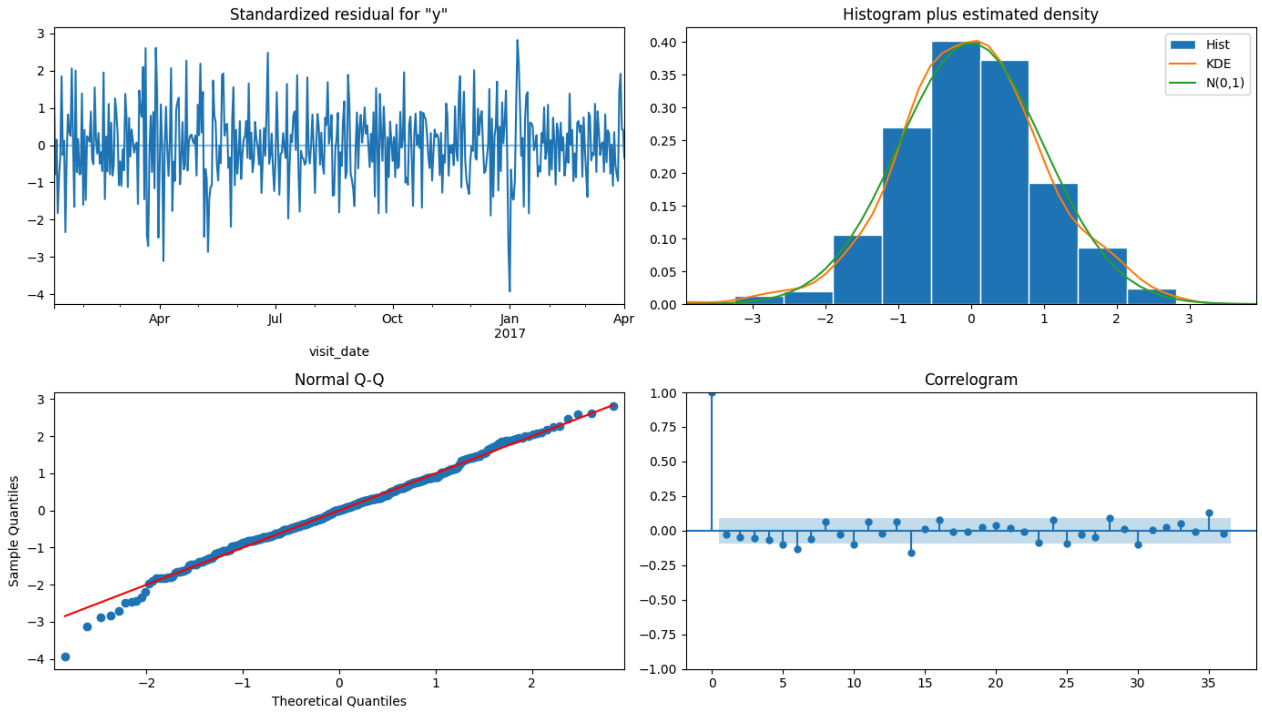
	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
ar.L1	-0.6220	0.040	-15.429	0.000	-0.701	-0.543
ar.L2	-0.5561	0.051	-10.990	0.000	-0.655	-0.457
ar.L3	-0.4072	0.053	-7.716	0.000	-0.511	-0.304
ar.L4	-0.2907	0.052	-5.562	0.000	-0.393	-0.188
ar.L5	-0.1930	0.042	-4.561	0.000	-0.276	-0.110
ar.S.L7	0.3951	0.039	10.196	0.000	0.319	0.471
ar.S.L14	0.2978	0.041	7.308	0.000	0.218	0.378
sigma2	2313.6353	142.278	16.261	0.000	2034.775	2592.495

Ljung-Box (L1) (Q):	0.26	**Jarque-Bera (JB):**	7.62
Prob(Q):	0.61	**Prob(JB):**	0.02
Heteroskedasticity (H):	0.68	**Skew:**	-0.12
Prob(H) (two-sided):	0.02	**Kurtosis:**	3.59

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

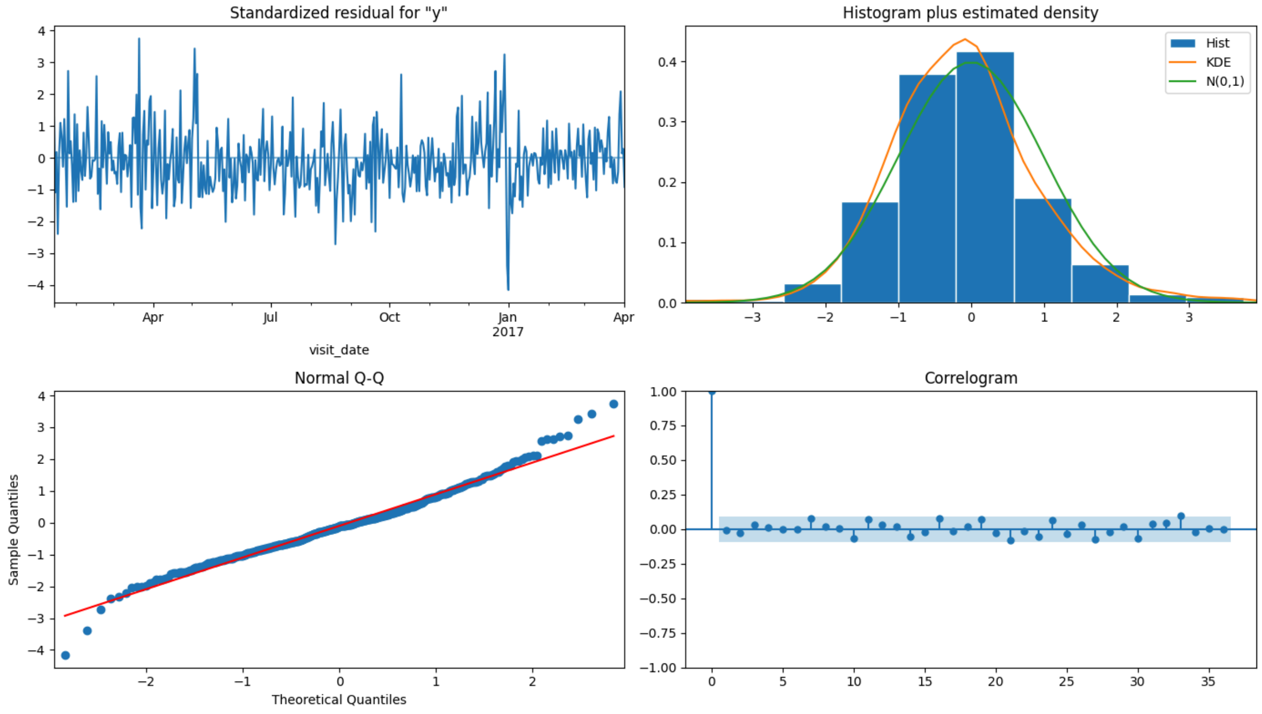
In [68]:

```
best_model_auto = ((5,1,0),(2,0,0,7))
fit_best_model_auto = SARIMAX(train, order=best_model_auto[0], seasonal_order=best_model_auto[1]).fit()
fit_best_model_auto.plot_diagnostics(figsize=(14, 8),lags=36)
plt.tight_layout()
```



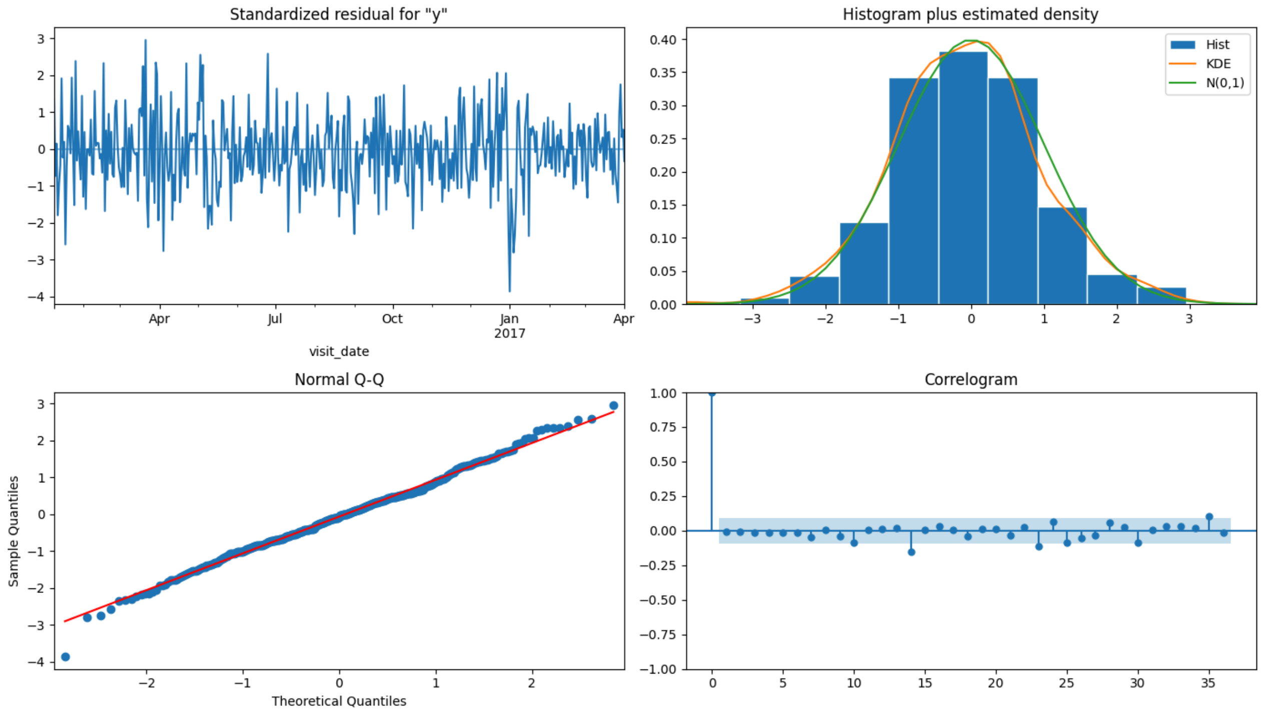
In [69]:

```
best_model_eye = ((1,0,1),(0,1,1,7))
fit_best_model_eye = SARIMAX(train, order=best_model_eye[0], seasonal_order=best_model_eye[1]).fit()
fit_best_model_eye.plot_diagnostics(figsize=(14, 8),lags=36)
plt.tight_layout()
```



In [70]:

```
best_model_grid = ((3, 1, 3), (2, 0, 0, 7))
fit_best_model_grid = SARIMAX(train, order=best_model_grid[0], seasonal_order=best_model_grid[1]).fit()
fit_best_model_grid.plot_diagnostics(figsize=(14, 8),lags=36)
plt.tight_layout()
```



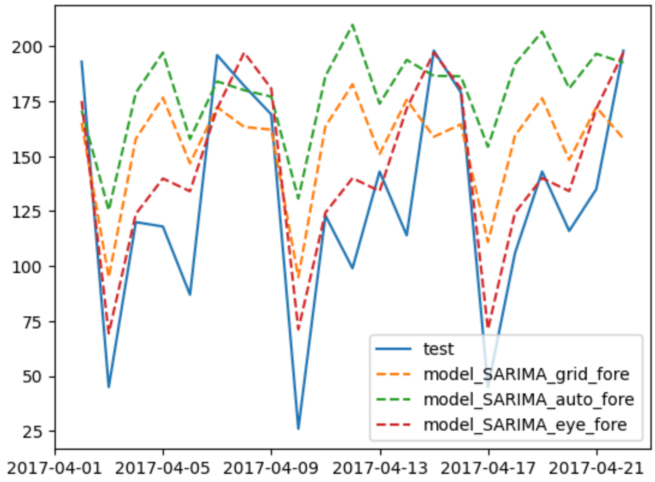
In [71]:

```
model_SARIMA_grid_fore = fit_best_model_grid.forecast(len(test))
model_SARIMA_auto_fore = fit_best_model_auto.forecast(len(test))
model_SARIMA_eye_fore = fit_best_model_eye.forecast(len(test))

plt.plot(test, label='test')
plt.plot(model_SARIMA_grid_fore,'--', label='model_SARIMA_grid_fore')
plt.plot(model_SARIMA_auto_fore,'--', label='model_SARIMA_auto_fore')
plt.plot(model_SARIMA_eye_fore,'--', label='model_SARIMA_eye_fore')
# plt.plot(test_fore_seas,'--', label='model_seas_fit')
plt.legend()
```

Out[71]:

```
<matplotlib.legend.Legend at 0x21e41419d60>
```



In [76]:

```
from statsmodels.tools.eval_measures import mse
print('MSE test (model_ARMA_fit_eye) %s:' % str(best_model_eye), mse(model_SARIMA_eye_fore,test)**0.5)
print('MSE test (model_ARMA_fit_auto) %s:' % str(best_model_auto), mse(model_SARIMA_auto_fore,test)**0.5)
print('MSE test (model_ARMA_fit_grid) %s:' % str(best_model_grid), mse(model_SARIMA_grid_fore,test)**0.5)
```

```
MSE test (model_ARMA_fit_eye) ((1, 0, 1), (0, 1, 1, 7)): 26.19714875040627
MSE test (model_ARMA_fit_auto) ((5, 1, 0), (2, 0, 0, 7)): 64.75426971263295
MSE test (model_ARMA_fit_grid) ((3, 1, 3), (2, 0, 0, 7)): 45.790202230346964
```

In [86]:

```
train = train.rename('visitors')
test = test.rename('visitors')
```

In [87]:

```
file = 'date_info'
holidays = pd.read_csv('data/restaurant/%s.csv' % file)
holidays.set_index('calendar_date',drop=True, inplace=True)
holidays.index = pd.to_datetime(holidays.index)
train_new = pd.concat([train, holidays['holiday_flg']], axis=1).dropna()
test_new = pd.concat([test, holidays['holiday_flg']], axis=1).dropna()
```

In [93]:

```
train_new
```

Out[93]:

	visitors	holiday_flg	is_weekend
2016-01-08	167.0	0	0
2016-01-09	293.0	0	1
2016-01-10	198.0	0	1
2016-01-11	210.0	1	0
2016-01-12	102.0	0	0
...	...	...	...
2017-03-28	168.0	0	0
2017-03-29	225.0	0	0
2017-03-30	158.0	0	0
2017-03-31	189.0	0	0
2017-04-01	168.0	0	1

450 rows × 3 columns

In [89]:

```
train_new["is_weekend"] = 1*(train_new.index.weekday >= 5)
test_new["is_weekend"] = 1*(test_new.index.weekday >= 5)
test_new
```

Out[89]:

	visitors	holiday_flg	is_weekend
2017-04-02	193.0	0	1
2017-04-03	45.0	0	0
2017-04-04	120.0	0	0
2017-04-05	118.0	0	0
2017-04-06	87.0	0	0
2017-04-07	196.0	0	0
2017-04-08	182.0	0	1
2017-04-09	169.0	0	1
2017-04-10	26.0	0	0
2017-04-11	123.0	0	0
2017-04-12	99.0	0	0
2017-04-13	143.0	0	0
2017-04-14	114.0	0	0
2017-04-15	198.0	0	1
2017-04-16	179.0	0	1
2017-04-17	45.0	0	0
2017-04-18	106.0	0	0
2017-04-19	143.0	0	0
2017-04-20	116.0	0	0
2017-04-21	135.0	0	0
2017-04-22	198.0	0	1

In [90]:

```
fit_model_SARIMA_eye_exog = SARIMAX(train_new['visitors'],exog=train_new['holiday_flg'],order=best_model_eye[0],seasonal_order=best_model_eye[1])
#,enforce_invertibility=False)
# fit_model_ARMA_eye_exog.summary()
```

In [98]:

```
train_new.index.name = 'date'
train_new.to_csv('data/restaurant/train_NEW_restaurant_visitors.csv')
```

In [99]:

```
train_new
```

Out[99]:

	visitors	holiday_flg	is_weekend
date			
2016-01-08	167.0	0	0
2016-01-09	293.0	0	1
2016-01-10	198.0	0	1
2016-01-11	210.0	1	0
2016-01-12	102.0	0	0
...	...	...	...
2017-03-28	168.0	0	0
2017-03-29	225.0	0	0
2017-03-30	158.0	0	0
2017-03-31	189.0	0	0
2017-04-01	168.0	0	1

450 rows × 3 columns

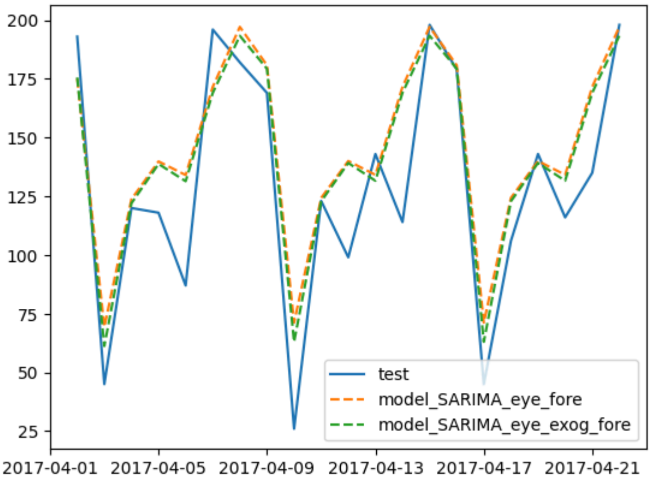
In [91]:

```
model_SARIMA_eye_exog_fore = fit_model_SARIMA_eye_exog.forecast(len(test),exog=test_new['holiday_flg'])

plt.plot(test, label='test')
plt.plot(model_SARIMA_eye_fore,'--', label='model_SARIMA_eye_fore')
plt.plot(model_SARIMA_eye_exog_fore,'--', label='model_SARIMA_eye_exog_fore')
# plt.plot(test_fore_seas,'--', label='model_seas_fit')
plt.legend()
```

Out[91]:

```
<matplotlib.legend.Legend at 0x21e415f9d60>
```



In [92]:

```
print('MSE test (model_ARMA_fit_eye) %s:' % str(best_model_eye), mse(model_SARIMA_eye_fore,test)**0.5)
print('MSE test (model_SARIMA_eye_exog_fore) %s:' % str(best_model_eye), mse(model_SARIMA_eye_exog_fore,test)**0.5)
```

```
MSE test (model_ARMA_fit_eye) ((1, 0, 1), (0, 1, 1, 7)): 26.19714875040627
MSE test (model_SARIMA_eye_exog_fore) ((1, 0, 1), (0, 1, 1, 7)): 24.060826616758806
```